



TRABAJO DE INVESTIGACIÓN

Technology–Enhanced Learning y Model–Driven Development — Revisión Sistemática de la Literatura

Autor:

Francisco J. Tena Rastrollo
University of Cádiz, Escuela Superior de Ingeniería,
C/ Chile 1, Cádiz 11002, Spain,
`francisco.tenarastrollo@alum.uca.es`

Tutores:

Juan Manuel Dodero Beardo, Manuel Palomo Duarte
University of Cádiz, Escuela Superior de Ingeniería,
C/ Chile 1, Cádiz 11002, Spain.

16 de diciembre de 2012

Licencia

Copyright © 2012 Francisco J. Tena Rastrollo. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

*“If I have seen further, it is by
standing on the shoulders of giants”*
— Isaac Newton

*“Dadme un punto de apoyo y
moveré el mundo”*
— Arquímedes

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Objetivo	2
1.3. Método de investigación	2
1.4. Estructura del documento	3
2. Desarrollo Dirigido por Modelos	4
2.1. MDA	4
2.2. MDE & Eclipse	6
2.2.1. Un ejemplo sencillo	8
2.2.2. Más información	9
2.3. DSL — Domain Specific Language	9
2.3.1. ¿Qué son DSLs?	9
2.3.2. Ampliar conocimientos	11
3. TEL — Technology–Enhanced Learning	12
3.1. Definición	12
3.2. Diseño	13
3.2.1. Diseño basado en la investigación	13
3.2.2. Patrones de diseño	14
4. LMS — Learning Management System	16
4.1. Definición y listado de algunos LMS	16
4.2. Evolución de las plataformas	17
5. Revisión Sistemática de la Literatura — Método de investigación	20
5.1. Planear	20
5.1.1. Cuestiones realizadas	21
5.1.2. Fuente de datos y cadenas de búsqueda	21
5.1.3. Criterio de inclusión y exclusión	23
5.1.4. Asegurar la calidad	23
5.1.5. Extracción de datos y análisis	24
5.1.6. Proceso para dirigir la revisión	25

5.2.	Ejecutar	27
5.2.1.	Búsquedas de estudios	27
5.2.2.	Selección y listado de estudios primarios	29
5.2.3.	Asegurar la calidad	32
5.2.4.	Otros artículos y estudios	35
5.3.	Analizar y Extraer datos	36
5.4.	Discusión	41
5.4.1.	PI1: ¿Qué es MDE, MDA, TEL y LMS?	41
5.4.2.	PI2: ¿Qué relación hay entre MDE (o MDA) y TEL?	42
5.4.3.	PI3: ¿Qué relación hay entre TEL y LMS?	43
5.4.4.	PI4: ¿Qué aplicación tiene MDE (o MDA) en LMS?	44
6.	Conclusiones y trabajos futuros	45
7.	Conceptos	48
7.1.	Glosario genérico	48
7.2.	Glosario específico de Eclipse	50
	Bibliografía	52
	GNU Free Documentation License	62
1.	APPLICABILITY AND DEFINITIONS	62
2.	VERBATIM COPYING	64
3.	COPYING IN QUANTITY	64
4.	MODIFICATIONS	65
5.	COMBINING DOCUMENTS	67
6.	COLLECTIONS OF DOCUMENTS	67
7.	AGGREGATION WITH INDEPENDENT WORKS	67
8.	TRANSLATION	68
9.	TERMINATION	68
10.	FUTURE REVISIONS OF THIS LICENSE	69
11.	RELICENSING	69

Índice de figuras

1.1. Proceso de Revisión Sistemática de la Literatura	3
2.1. MDA Metamodel Description	5
2.2. MDA aplicado al diseño de sistemas TEL [27]	7
2.3. EMF. Modelo Ecore	9
2.4. EMF. Estructura del Diagrama Ecore	10
4.1. Generación de los LMS	19
5.1. Proceso para dirigir la revisión	25

Índice de cuadros

5.1. Cadenas de búsquedas realizadas	22
5.2. Resultados de las búsquedas desglosados	27
5.3. Resultados totales de las búsquedas	28
5.4. Listado de estudios primarios de IEEEExplore	29
5.5. Listado de estudios primarios de Science Direct	30
5.6. Listado de estudios primarios de SpringerLink	31
5.7. Puntuación de calidad para los estudios primarios	33
5.8. Otros estudios	35
5.9. Análisis y Extracción de datos – parte 1	37
5.10. Análisis y Extracción de datos – parte 2	39

Capítulo 1

Introducción

1.1. Motivación

Durante el periodo formativo “Modelado, Simulación y Pruebas de Procesos y Tratamiento de Señales y de Datos”, tanto yo como otros muchos alumnos, fuimos observando y fijándonos en qué ofrecían los profesores como temas de investigación. Por desgracia, a pesar de que muchas de las propuestas eran sumamente interesantes, había que quedarse con una.

En mi caso, que siempre me ha gustado todo lo relacionado con los procesos de Ingeniería del Software y el desarrollo software en general, me decanté por aprender de Juan Manuel Doderó, Manuel Palomo e Ivan Ruíz lo que se conoce como la ingeniería dirigida por modelos, o en inglés *Model-Driven Engineering*, cuyo acrónimo es MDE.

Muy resumidamente, pues a lo largo de este documento se aprenderá el concepto en profundidad, MDE es una metodología de desarrollo software que se centra en crear y explotar modelos de dominio (representaciones abstractas del conocimiento y actividades que gobiernen un dominio en particular), más que en conceptos algorítmicos o computacionales (como la ingeniería clásica) [94]. La ingeniería dirigida por modelos ofrece una aproximación para llegar a suplir la incapacidad o la falta de potencia de los lenguajes de tercera generación, de modo que pueda aliviarse o de algún modo mitigarse la complejidad de las plataformas actuales y así poder expresar con eficacia conceptos de dominio [94].

Lo cierto es que el significado y la utilidad que tiene MDE me llamó poderosamente la atención. Cuando empecé el curso de Doctorado, desconocía todo este universo. Sin lugar a dudas, hay mucho que aprender y mucho por descubrir. Todo este campo abierto que la ingeniería dirigida por modelos es, en resumen, la motivación de la que surge este Trabajo de Investigación.

1.2. Objetivo

Dado que usando MDE se puede modelar cualquier concepto de la vida real, y debido a que en el grupo de investigación Mejora del Proceso Software y Métodos Formales (SPI&FM, o *Software Process Improvement and Formal Methods* en inglés) es un punto importante dentro de los estudios que llevan a cabo, MDE y el aprendizaje potenciado o basado en la tecnología (en inglés, *Technology-Enhanced Learning*, o su acrónimo TEL) son elementos que se desarrollaran a lo largo del Trabajo de Investigación como si fueran uno sólo, combinados para así entender lo que MDE puede aportar.

Dicho esto, los objetivos de este trabajo son básicamente tres:

Conceptos Describir los distintos conceptos que rodean a MDE y TEL.

Estado actual Presentar el estado actual de esta metodología, pero no por si sola, que tampoco llevaría a descubrir mucho más de lo ya conocido hoy día, sino aplicada y combinada, principalmente, al ámbito de la enseñanza y el aprendizaje. A pesar de que mis investigaciones empezaron por MDE, luego derivaron en todo lo que se verá a lo largo del documento.

Revisión sistemática de la literatura Para llegar el punto anterior, ha sido necesario seguir un detallado método de investigación (ver la sección 1.3) y realizar la correspondiente revisión de la literatura.

Cabe destacar que todos los conceptos que aquí se tratarán son muy recientes. Tras un análisis sistemático de la bibliografía sobre el tema, se observa que los documentos, artículos, libros y reseñas encontrados hasta ahora suelen datar del 2008 o 2009 en adelante [60], siendo durante los años 2011 y 2012 cuando la información se vuelve más precisa y, por decirlo de forma sencilla, ahondan con mayor precisión en el aprendizaje apoyado en la ingeniería dirigida o conducida por modelos.

1.3. Método de investigación

El método de investigación utilizado en este estudio es una revisión sistemática de la literatura basada en las guías propuestas por [51] y [11].

La figura 1.1 de la página 3 muestra gráficamente y a muy alto nivel cómo es el proceso descrito en ambos artículos.

Siguiendo estas dos guías, el proceso de revisión sistemática de la literatura se ha compuesto de tres fases consecutivas: *planear*, *ejecutar* y *analizar* los resultados. Existe, además, otra fase que llamamos *empaquetado*, que se realiza a través y durante el proceso completo para poder almacenar los resultados de las fases previas. Finalmente, se ejecutan dos chequeos en el transcurso del proceso para evaluar que el proceso de la revisión sistemática de la literatura está siendo el correcto [11].

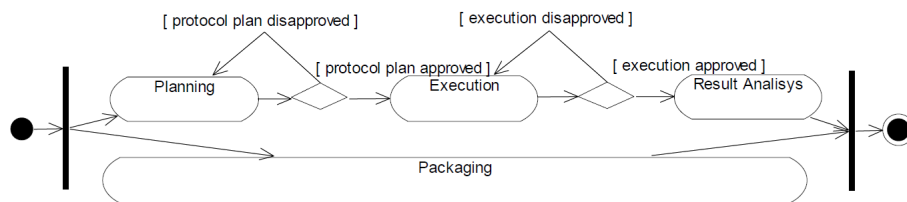


Figura 1.1: Proceso de Revisión Sistemática de la Literatura [11]

1.4. Estructura del documento

Este Trabajo de Investigación empieza explicando los conceptos necesarios para abordar no sólo la posterior revisión sistemática de la literatura, sino también para conocer para qué sirven, qué utilidad tienen hoy día y para descubrir cuál de ellos servirán o no en futuras investigaciones.

El capítulo 2 aborda el Desarrollo Dirigido por Modelos, y más concretamente MDA, MDE y los DSL.

El capítulo 3 explica qué es TEL (*Technology-Enhanced Learning*, en inglés) y su diseño dentro del ámbito de la enseñanza y el aprendizaje.

El capítulo 4 presenta los sistemas de gestión de aprendizaje, o LMS, que están íntimamente relacionados con TEL además de con el desarrollo dirigido por modelos.

Posteriormente, se realiza, en el capítulo 5, la revisión sistemática de la literatura propuesto por B. Kitchenham [51]. Es un capítulo amplio que termina dando respuesta a una serie de preguntas planteadas.

En el capítulo 6 se exponen las conclusiones y trabajos futuros.

Se ha considerado además que es muy conveniente conocer al menos la terminología utilizada a lo largo de trabajo. Por este motivo, se ha confeccionado un glosario en el capítulo 7.

Finalmente, se lista la bibliografía y la licencia utilizada.

Capítulo 2

Desarrollo Dirigido por Modelos

2.1. MDA

Hablar de MDE es hacerlo, a su vez, de MDA (*Model-Driven Architecture*, en inglés). Ambos términos son de reciente aparición. MDA, por su lado, es un estándar de modelo-dirigido propuesto por el OMG (*Object Management Group*), organización conocida por su especificación de CORBA y UML.

El estándar MDA facilita la definición de guías, ayudando en la creación de los diferentes pasos del diseño. MDA también facilita el soporte para las herramientas utilizadas en el desarrollo de sistemas de la información, las cuales afectan directamente a la productividad y al mantenimiento de dichos sistemas [81].

MDA presenta la ventaja de separar aspectos de diseño de los temas relacionados con la arquitectura. Para este fin, define tres niveles de abstracción, relativamente similares a la distinción tradicional entre los niveles conceptuales, lógicos y físicos en la ingeniería de sistemas de la información.

- *Computation Independent Model* (CIM)
- *Platform Independent Model* (PIM)
- *Platform Specific Model* (PSM)

CIM constituye el nivel más abstracto. Representa el contexto y el objetivo del sistema de la información sin ninguna consideración computacional. Se centra en las consideraciones conceptuales, de negocio [81] o del dominio del modelo [76]. Por lo general, el modelo es independiente de cómo el sistema es implementado.

PIM describe el comportamiento y estructura del sistema de la información, sin tener en cuenta las elecciones en cuanto a la implementación [81]. Es por

tanto un modelo que será aplicable a un estilo de arquitectura particular, o quizá varios.

Finalmente, el tercer nivel, PSM, contiene toda la información necesaria que permite a los desarrolladores construir el código y ejecutar la aplicación resultante [81]. Es tarea del arquitecto software el elegir la plataforma en función del sistema que se desea construir [76].

Aplicar MDA al desarrollo de los sistemas de conocimiento requiere un mapeo entre los niveles de abstracción de MDA y los modelos de conocimiento. Estos mapeos se pueden hacer usando varias herramientas o *tools* [81]. Un mapeo de MDA provee de especificaciones para transformar un PIM en PSM para una plataforma o sistema particular. El modelo de dicha plataforma determinará la naturaleza del mapeo [76].

La figura 2.1 de la página 5 nos da una idea de cómo MDA describe los mapeos. PIM, PSM y las técnicas de mapeo están basadas en meta-modelos expresados usando las tecnologías propias de OMG, como UML, MOF y CWM.

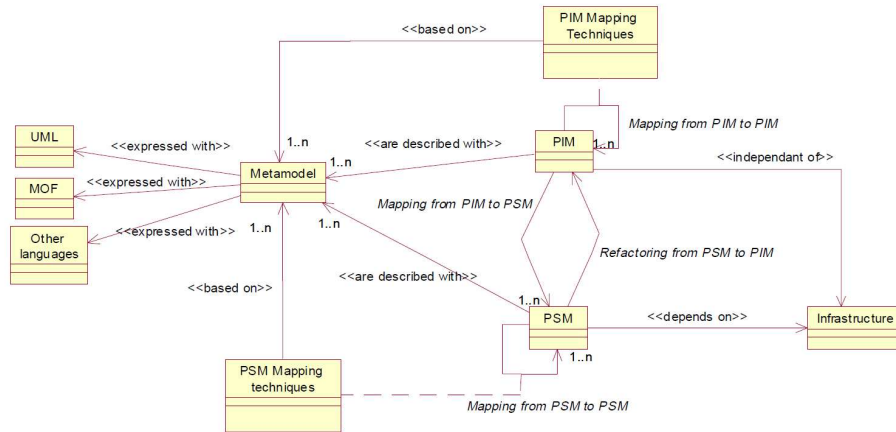


Figura 2.1: MDA Metamodel Description

Después de todo lo dicho, puede observarse que MDA tiene todo el potencial necesario para proveer una alta rentabilidad a las compañías que se decanten por este tipo de desarrollos. Entre dichos beneficios podemos mencionar la portabilidad, interoperabilidad y la reusabilidad [115].

MDA se centra principalmente en el desarrollo de aplicaciones transaccionales. Sin embargo, es aplicable al desarrollo de otro tipo de aplicaciones. MDA, y en general cualquier sistema modelo-dirigido, provee de grandes beneficios al desarrollo de almacenes de datos [66]. Pero también MDA potencia en gran medida los desarrollos de sistemas basados en reglas y, más genéricamente, pero de hecho lo más importante para este estudio, los desarrollos relacionados con la ingeniería del conocimiento.

Gracias a que MDA distingue entre tres niveles de abstracción [81], después de definir un modelo en el nivel PIM, los desarrolladores pueden generar los

modelos que quieran en el nivel PSM, dependiendo del sistema que necesiten implementar.

Aunque se hablará de TEL (*Technology-Enhanced Learning*) más adelante (ver capítulo 3), se quiere aprovechar esta sección para mencionar que MDA es totalmente válido como proceso de diseño de TEL [27].

Siguiendo las prácticas de MDA, un escenario de diseño típico basado en MDA puede describirse del siguiente modo:

Primero, el diseñador define informalmente el escenario de aprendizaje que se quiere crear y los recursos necesarios (CIM).

Para formalizar el modelo pedagógico, se utiliza un lenguaje de modelado. Este lenguaje permite definir el método pedagógico y las restricciones del mismo. En términos de MDA, el modelo pedagógico sería un PIM.

Finalmente, para obtener un sistema que pueda ser ejecutado por un LMS (ver capítulo 4), el modelo pedagógico debe ser transformado en un modelo técnico (PSM). Esto corresponde al metamodelo LMS. La forma más común de transformar un modelo en otro es utilizar un motor de transformación de modelo (como ATL [27]) y un conjunto de reglas dedicadas a este tipo de mapeo.

La figura 2.2 de la página 7 representa de forma gráfica lo explicado anteriormente y cómo MDA ayuda al proceso seguido para diseñar un TEL.

Se ha mencionado que la diferencia entre un concepto pedagógico (relacionado con PIM) y el técnico (relacionado con PSM) consiste en el hecho de que el concepto pedagógico es abstracto y está relacionado con los intereses pedagógicos del diseñador, pero el concepto técnico es concreto y representa una herramienta que permite la ejecución del correspondiente concepto pedagógico dentro del LMS. De hecho, los conceptos pedagógicos permanecen abstractos hasta que se contextualizan en términos del LMS en el paso dedicado a la transformación [27].

2.2. MDE & Eclipse

Relacionar Eclipse con MDE (*Model-Driven Engineering*, en inglés) es “bajar” de nivel, pero un término va asociado al otro. Se puede considerar MDA como simplemente una encarnación específica de la aproximación a la ingeniería dirigida por modelos aplicada a los sistemas software. Pero MDE no está restringido al estándar MDA. De hecho, MDE puede no estar restringido al desarrollo y evolución de los sistemas software [29]. MDE es abierto, al contrario que MDA, que lo es bastante menos. MDE tiene más campo de acción que MDA y combina *proceso* y *análisis* con arquitectura [50].

A diferencia de MDA, que cubre el ciclo completo de un desarrollo y la integración de las aplicaciones, MDE usa lo que Eclipse denomina EMF, que se centra en el modelado de clases.

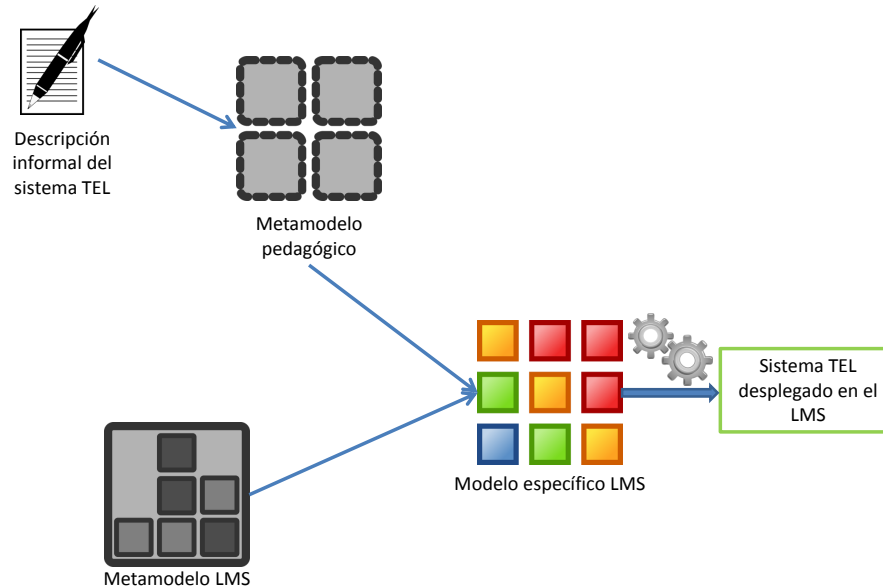


Figura 2.2: MDA aplicado al diseño de sistemas TEL [27]

Eclipse ofrece numerosos *frameworks*, y el que se utiliza para el modelado de datos es EMF. Eclipse Modeling Framework es un *framework* en Java/XML para generación de herramientas y otras aplicaciones basadas en modelos de clases simples. EMF ayuda a transformar modelos en código Java personalizable, correcto y eficiente. Está pensado para proveer los beneficios del modelado formal, pero con un bajo coste de entrada o inicio. Provee además la posibilidad de salvar objetos como documentos XML para intercambiarlos con otras herramientas y aplicaciones. Los modelos pueden ser creados usando Java, XML o herramientas de modelado como Rational Rose y UML, y posteriormente importarlos a EMF. Bien es cierto que Eclipse y EMF ofrecen su propio entorno para el diseño de modelos [28].

El generador de código convierte un modelo en un conjunto de clases en implementación Java. Estas clases son extensibles y regenerables, de modo que pueden modificarse añadiendo métodos y variables definidos por el usuario. Si el modelo cambia, pueden regenerarse las clases de implementación, manteniéndose las modificaciones realizadas. Esto funciona en dos sentidos. Es decir, que cambios en el código Java pueden a su vez utilizarse para actualizar el modelo [28].

Es importante recordar que, cuando decimos “modelo”, estamos pensando en Diagramas de Clases, Diagramas de Colaboración, Diagramas de Estado y

demás elementos que conforman un diseño software. UML define un (o debiéramos decir **el**) estándar de notación para ese tipo de diagramas. Usando una combinación de diagramas UML, se puede especificar el modelo completo de una aplicación [28].

2.2.1. Un ejemplo sencillo

Para entender mejor lo que Eclipse y EMF puede hacer por nosotros y ofrecernos, vamos a mostrar un sencillo ejemplo.

EMF distingue entre el meta-modelo y el modelo actual. El meta-modelo describe la estructura del modelo. Un modelo es por tanto la instancia de un determinado meta-modelo. EMF provee un *framework* basado en *plug-ins* para almacenar la información del modelo. El sistema por defecto para guardar esta información es XMI [109].

Aunque EMF permite crear el meta-modelo de varias formas (XMI, Java, UML o esquemas XML), lo más sencillo es usar directamente las herramientas que ofrece EMF a través de Eclipse [109].

Como hemos dicho, EMF se basa en meta-modelos. Concretamente en dos de ellos: Ecore y Genmodel. El meta-modelo Ecore contiene la información sobre la definición de las clases. Genmodel, por su lado, contiene información adicional para la generación de código (por ejemplo, rutas, información de ficheros, parámetros sobre cómo debería generarse el código) [109].

Ecore permite definir los siguientes elementos:

- *EClass*, que representa una clase con atributos y referencias.
- *EAttribute*, que representa un atributo con nombre y tipo de datos.
- *EReference*, que representa asociaciones o referencias entre dos clases.
- *EDataType*, que representa el tipo de un atributo, como por ejemplo un entero, un flotante o cualquier tipo de datos de Java, como puede ser *java.util.Date*.

El primer paso, por tanto, es crear un modelo Ecore. Si estuviéramos modelando una página web sencilla, el modelo sería el que podemos ver en la figura 2.3 de la página 9.

El diagrama anterior es la visualización de cómo queda el modelo usando la herramienta Eclipse, pero a nivel estructural, podemos ver el resultado en la figura 2.4 de la página 10.

Una vez llegado a este punto, puede obtenerse a través de un generador de modelos (el anteriormente citado Genmodel) el código Java, que contiene lo siguiente:

- *model*: interfaces y la Factoría para crear clases Java.
- *model.impl*: implementación concreta de las interfaces definidas en el modelo.

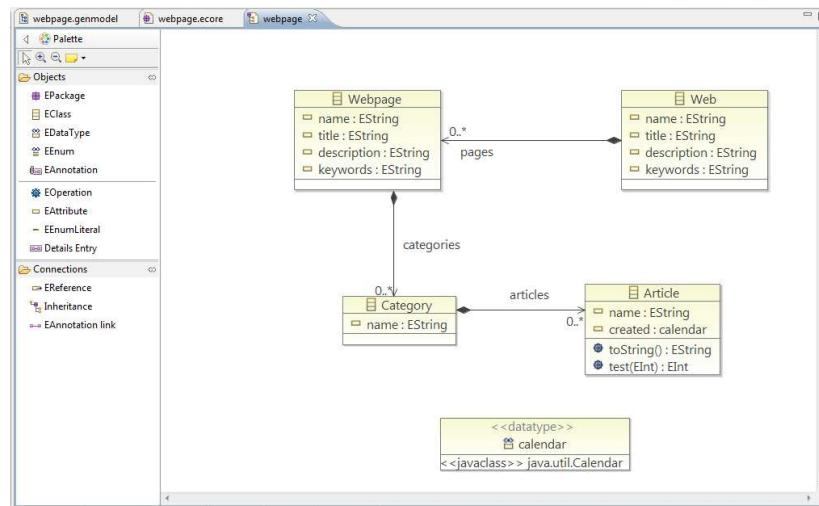


Figura 2.3: EMF. Modelo Ecore

- *model.util*: factoría Adapter que ofrece una serie de métodos para los modelos.

Todo el código puede modificarse libremente, así como ser usado para generar otros objetos y métodos. También pueden generarse *plug-ins* para usar el modelo en otro diferente a modo de paquete externo, documentación automática JavaDoc o extender, mediante herencia, el modelo Ecore [109].

2.2.2. Más información

Durante la revisión sistemática de la literatura realizada en este Trabajo de Investigación se encontró un artículo que puede servir para completar lo que se ha redactado en este trabajo. Se invita al lector a acudir a dicho artículo, titulado “Application of model-driven engineering (MDA) for the construction of a tool for domain-specific modeling (DSM) and the creation of modules in learning management systems (LMS) platform independent” [60].

2.3. DSL — Domain Specific Language

2.3.1. ¿Qué son DSLs?

Un lenguaje de dominio específico o DSL (en inglés, *domain-specific language*) es un “mini” lenguaje construido sobre un lenguaje base que provee una sintaxis y semántica común para representar conceptos y comportamientos en un dominio particular o específico. Tanto DSL, MDA y MDE son conceptos relacionados de uno u otro modo.

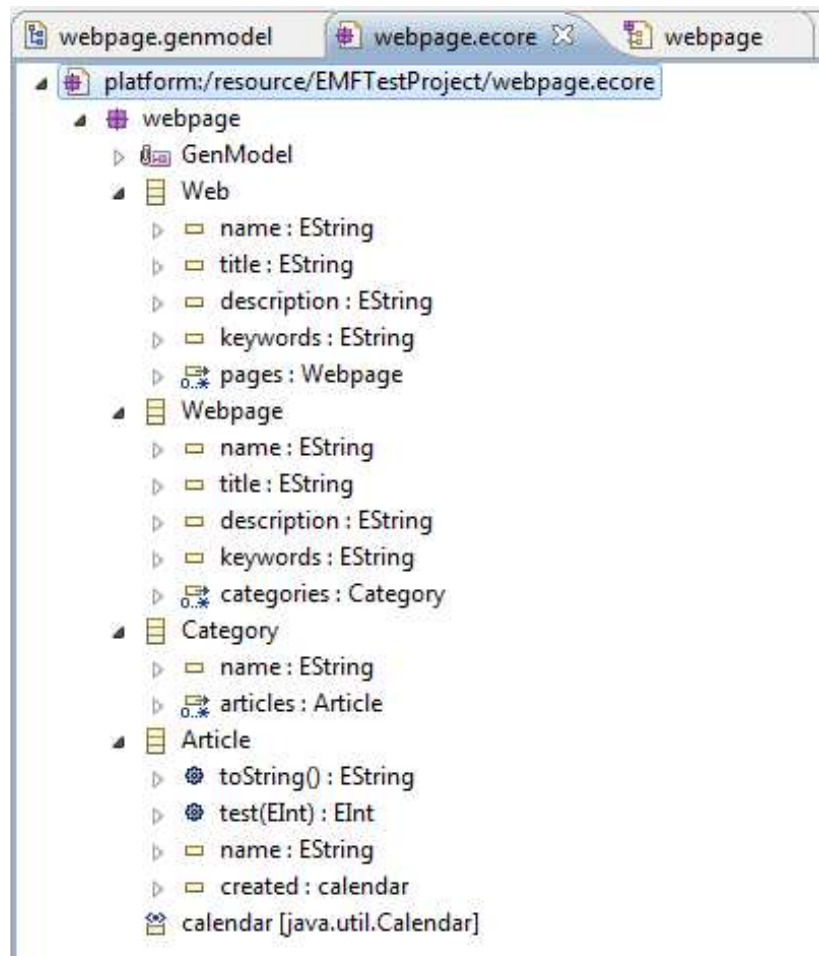


Figura 2.4: EMF. Estructura del Diagrama Ecore

En general, usar o diseñar un DSL ayuda a alcanzar los siguientes objetivos [65]:

1. Abstracción permitiendo la programación a un mayor nivel que el que está disponible usando el lenguaje de programación base o sus librerías. Un DSL permite a los conceptos del dominio, las acciones y los comportamientos ser representados directamente en la nueva sintaxis ideada.
2. Código efectivo y homogéneo, como efecto colateral de la programación en un alto nivel de abstracción.
3. Sintaxis natural y simple, que favorece una forma sencilla de escribir y leer código.

4. Sencillez en la programación, que es deseable en cualquier lenguaje de programación y también de algún modo difícil de juzgar. Sin embargo, ya que DSL permite la expresión de constructores que mapean directamente a un dominio específico, esto hace generalmente la programación más sencilla (para aplicaciones en el dominio) que usando directamente el lenguaje subyacente.
5. La generación de código es una de las funciones principales de DSL. Esencialmente, las sentencias DSL son traducidas en tiempo de ejecución al código que utiliza el lenguaje subyacente y sus librerías.

2.3.2. Ampliar conocimientos

Durante la revisión sistemática de la literatura realizada en este Trabajo de Investigación se encontró un artículo tremendamente interesante que trata en profundidad los DSL. Se invita al lector a acudir a dicho artículo, titulado “Domain-specific languages: an annotated bibliography” [22].

También se estudió otro artículo, que lleva por título “When and how to develop domain-specific languages” [67], el cual es de especial relevancia para cualquier desarrollador de DSL. Ahonda en las fases de desarrollo de los DSL: decisiones de análisis, diseño e implementación. También se discuten herramientas que pueden ayudar a un desarrollo más rápido de los DSL.

Capítulo 3

TEL — Technology–Enhanced Learning

3.1. Definición

Llegados a este punto, y conociendo lo que MDE nos ofrece, es interesante explicar qué es TEL y su relación con la ingeniería dirigida por modelos. Además, de este modo, entenderemos mejor la idea de los LMS, que trataremos más adelante (ver 4), y los DSL, que acaban de ser explicados (ver 2.3).

Hay muchas definiciones de qué es TEL, que podemos traducirlo como aprendizaje basado en la tecnología o aprendizaje potenciado por la tecnología.

Básicamente, TEL es el soporte a través de la tecnología de cualquier actividad relacionada con el aprendizaje. TEL, a menudo, se usa como sinónimo de e–Learning, a pesar de que son cosas muy diferentes. La principal diferencia entre estas dos expresiones es que TEL se centra en el soporte tecnológico de cualquier campo pedagógico que use la tecnología [114].

Una actividad relacionada con el aprendizaje puede describirse en términos de [114]:

- *recursos*: creación, distribución, acceso, compilación y consumo de contenido digital, así como herramientas y servicios.
- *acciones*: comunicación, colaboración e interacción con herramientas software.
- *contexto*: tiempo, duración y localización.
- *roles*: una actividad es llevada a cabo por varios actores en roles que pueden ir cambiando (por ejemplo, estudiante, profesor, entrenadores, gestores, etc.)

- *objetivo*: soporte en la consecución de los objetivos de aprendizaje, respetando las preferencias individuales y de organización o equipo.

Por tanto, una definición de qué es TEL sería la siguiente:

TEL tiene el objetivo de proveer innovaciones socio-tecnológicas (así como mejorar la eficiencia y la rentabilidad) en prácticas que tengan que ver con el aprendizaje, ya estén relacionadas con particulares u organizaciones, y que son independientes del tiempo, el lugar o el momento. El campo de actuación de TEL describe por tanto el soporte de cualquier actividad de aprendizaje a través de la tecnología.

Como observamos, TEL es algo muy amplio y abarca un terreno demasiado grande como para ser tratado al completo en este artículo. Durante la revisión sistemática de la literatura realizada en este Trabajo de Investigación se encontró un artículo tremendamente interesante que abarca TEL aplicado a la ciencia de la computación. Se invita al lector a acudir a dicho artículo, titulado “A Computer Science Perspective on Technology–Enhanced Learning Research. Technology–Enhanced Learning” [102].

Dado que nuestra intención es la relación de la ingeniería dirigida por modelos y TEL, vamos a describir a continuación cual es el camino a seguir cuando mezclamos diseño (en el sentido amplio de la palabra) y TEL.

3.2. Diseño

El diseño es un aspecto clave para el desarrollo exitoso de cualquier entorno de aprendizaje interactivo. De hecho, en TEL, el proceso de diseño requiere entradas de información de distintas áreas de experiencia [73].

Lo que se conoce como investigaciones basadas en el diseño es una metodología para el estudio de funciones. A menudo se le conoce como diseño de investigaciones o diseño de experimentos, pero en todo caso, está relacionado con el diseño de procesos de aprendizaje.

3.2.1. Diseño basado en la investigación

Centrándonos ya en el diseño de TEL, podemos listar los siguientes elementos que deben tenerse en cuenta para todo aprendizaje potenciado o basado en la tecnología [73]:

- El objetivo del diseño para procesos de aprendizaje debe ser desarrollar los distintos tipos de teorías que cubran dicho proceso.
- Los diseños para procesos de aprendizaje son diversos y altamente intervencionistas. Ya que normalmente tienen un fuerte componente de innovación, hay a menudo diferencias entre situaciones “típicas”, que pueden ser estudiadas de forma natural, y aquellos otros elementos que deben de realizarse mediante experimentos.

- Los diseños para procesos de aprendizaje tienen siempre dos caras: potenciales y reflexivos. Por el lado potencial, los diseños son implementados con procesos de aprendizaje hipotéticos y los mecanismos ideados quedan en la mente del diseñador. Por el lado reflexivo, los diseños se componen de diferentes *tests* dirigidos mediante conjeturas junto con diferentes niveles de análisis.
- Juntos los aspectos potenciales y reflexivos resultan en un diseño iterativo. Ya que las conjeturas son generadas y posteriormente probadas, y en ocasiones confirmadas o quizá refutadas, se deben desarrollar nuevas conjeturas sujetas a más pruebas.
- Las teorías desarrolladas durante el proceso de experimentación deben ser “modestas”, no en el sentido de ser específicas para un proceso de aprendizaje concreto y de un dominio específico, sino porque son las principales responsables en el diseño de TEL.

Los elementos de diseño pueden referirse a la pedagogía, a una actividad, o las herramientas utilizadas para el aprendizaje (ver 3.1). En algunos casos, los investigadores se centran en el refinamiento iterativo del diseño educativo mientras mantienen las herramientas “fijas”, mientras que en otros casos pueden centrarse en las herramientas, aplicando a las actividades un proceso de construcción menos estricto. Puede incluso considerarse un diseño coherente y válido el tener un sistema que lo contemple todo por igual.

3.2.2. Patrones de diseño

Un patrón de diseño describe un problema que ocurre una y otra vez en nuestro entorno, y dicho patrón aporta la solución principal a ese problema, de modo que se pueda usar esta solución todas las veces que sean necesarias sin repetir la misma manera de hacer las cosas dos veces [73].

Es una definición un tanto original, pero totalmente cierta. Sitúa a los patrones como una especificación de alto nivel, como método de solucionar un problema mediante un diseño que especifique el contexto del mismo, las particularidades de dicho problema, y cómo éstas pueden acometerse mediante los instrumentos de diseño.

Los patrones de diseño aplicados al aprendizaje tienen tres tendencias principales [73]:

- Patrones de diseño pedagógicos.
- El desarrollo de patrones de diseño software dedicados a crear tecnologías para la educación.
- La búsqueda de patrones en prácticas como la evaluación, valoraciones, análisis del aprendizaje y análisis de los sistemas dedicados al aprendizaje.

El primer punto, los patrones de diseño pedagógicos, se basan en que los profesionales experimentados en educación ya deben haber intentado y probado métodos para solucionar problemas recurrentes o afrontado necesidades comunes. Un buen ejemplo de esto puede encontrarse en¹, proyecto iniciado por un grupo de ingenieros software y educadores del campo de la ciencia de la computación [73]. Este proyecto propone un conjunto de patrones que abarcan temas desde el diseño de un curso hasta principios específicos de la ciencia de la computación, dando a su vez instrucciones para solucionar problemas concretos.

El segundo campo en el cual han proliferado los patrones de diseño aplicados a TEL es el de las tecnologías aplicadas a la educación, y especialmente las herramientas basadas en la web. Ejemplos notables en este campo incluyen el proyecto E-LEN² y numerosas iniciativas dentro del marco de trabajo de IMS-LD³. La mayoría de los trabajos en este área están centrados en los aspectos de diseño aplicados a la ingeniería software, y en el desarrollo, despliegue y elección de una buena tecnología para herramientas basadas en la web. Todo esto hace que este segundo enfoque sea bastante más técnico.

El tercer foco se basa principalmente en lo que se conoce como diseño participativo. Esto no es otra cosa que un conjunto de teorías, prácticas y estudios relacionados con el hecho de que los usuarios finales son los participantes que realmente harán uso de los productos software y el hardware así como de las actividades relacionadas con estos dos elementos. Como se observa, hay por tanto una relación estrecha entre aprendizaje y diseño. Un diseño efectivo debería poder construirse como un proceso de aprendizaje que involucrara a los usuarios y los diseñadores, y sólo generar un producto válido mediante el uso del producto inicialmente diseñado [73]. En resumen, esto no es más que un diseño iterativo que permite a los usuarios y diseñadores colaborar y pulir los conceptos del producto diseñado.

¹<http://pedagogicalpatterns.org/>

²<http://www2.tisip.no/E-LEN/>

³<http://www.imsglobal.org/>

Capítulo 4

LMS — Learning Management System

4.1. Definición y listado de algunos LMS

¿Qué es LMS? La traducción del inglés para este acrónimo es de por sí descriptivo: sistemas de gestión de aprendizaje. Básicamente son un conjunto de aplicaciones software orientadas a la formación de instituciones (universidades, por ejemplo) u organizaciones (como pueden ser empresas). Se podría decir que un LMS es la tecnología aplicada a TEL.

Hoy día, dentro de la sociedad tan cambiante en la que estamos inmersos, es necesario aprender constantemente e ir adquiriendo continuamente nuevos conocimientos. Los LMS permiten un aprendizaje autónomo.

Hay multitud de LMS actualmente, siendo los más utilizados aquello que son *open source*. Se listan a continuación los más conocidos:

- Docebo. Plataforma LMS utilizada en la enseñanza superior (universidades) y en la empresa. Soporta 18 idiomas y diferentes modelos didácticos, como Blended, Self-Directed, Collaborative e incluso aprendizaje social mediante chat, wikis y foros. Soporta el estándar SCORM.
- Moodle. Es un CMS (*Course Management System*). Es una aplicación web gratuita que permite crear a su vez entornos web completos dedicados al aprendizaje. Numerosas instituciones utilizan Moodle como plataformas para crear cursos online completos, aunque sin duda puede usarse también para cursos pequeños e incluso para cursos uno-a-uno. Dispone de foros, wikis y bases de datos para alojar cualquier tipo de contenido.
- Dokeos. Es una de las compañías más grandes dedicadas a los LMS *open source*. Su principal producto cumple con el estándar SCORM usado por multinacionales, administraciones y universidades. EL código está escrito en PHP y utiliza MySQL como base de datos. Permite todo tipo de

tests, grandes opciones para la interacción entre usuarios, alojamiento de presentaciones y algo más de 30 idiomas.

- Claroline. Plataforma colaborativa de e-learning y e-working lanzaba bajo licencia *open source* (GPL). Permite a cientos de organizaciones (universidades, escuelas, empresas, etc.) crear y administrar cursos y crear espacios de colaboración en la web. Plataforma usada en más de 80 países y disponible en más de 30 lenguajes. Compatible con GNU/Linux, Mac y Windows. Código basado en PHP y MySQL. La plataforma está basada en el concepto de “espacio” asociado a un curso o actividad pedagógica. Cada “espacio” provee de un conjunto de herramientas como descripción de cursos, publicación de documentos en cualquier formato, administración de foros, desarrollos, consultar estadísticas de estudiantes o wikis colaborativas.
- ATutor. Es un LMCS (*Learning Content Management System*) *open source* basado en la web. Suele citarse como el único con características de accesibilidad (útil por tanto para discapacitados visuales) y por su idoneidad para el uso educativo de acuerdo al criterio de evaluación establecido por el ASTD (*American Society for Training and Development*). ATutor cumple completamente con los estándares de XHTML con el objetivo de presentarse correctamente en cualquier tipo de tecnología compatible con dicho estándar. Es posible adaptarlo además fácilmente a una gran variedad de dispositivos como *smart phones*, PDAs, navegadores web o tabletas.
- Ilias. LMS *open source* que soporta gestión de contenidos basados en la web y un gran conjunto de herramientas para comunicación, foros, colaboración, evaluación, creación de ejercicios, etc. Publicado bajo *GNU General Public License*, se ejecuta sobre servidores PHP con MySQL. Cumple con el estándar SCORM y entre sus características principales se encuentran la posibilidad de tener espacios personales e individuales, gestión de grupos y cursos midiendo el progreso del aprendizaje, control de acceso basado en roles, manejo de formatos como XML, uso de chat, foro y varios tipos de *test*, alojamiento de *podcasts* e incluso soporte para *Google Maps*.

Estos y otros LMS son descritos ampliamente en varios de los artículos recogidos durante la revisión sistemática de la literatura, como en las referencias [87] y [106].

4.2. Evolución de las plataformas

Los sistemas de gestión de aprendizaje o LMS han dominado la educación basada en la web durante las pasadas dos décadas [20]. Pero la tecnología avanza rápido, y estos sistemas han sufrido y sufrirán modificaciones para adaptarse a las nuevas tecnologías y ser más dinámicos y menos “pasivos”.

Ya se ha hablado de las iniciativas *open source*. Las hay también propietarias, como WebCT/Blackboard, Graderpoint, Desire2Learn y Learn.com. En

todo caso, todas ellas suelen estar construidas sobre *frameworks* extensibles que permite ajustar y modificar las necesidades específicas.

La **primera generación** de plataformas e-learning (sobre el 1993) eran, en esencia, soluciones cerradas. Un sistema, un determinado curso. Estos sistemas se centraban en cursos particulares y contenido diseñado para un único fin. Durante esta época, surgen estándares como Dublin Core¹, IMS Learning Resource Metadata², IEEE Learning Object Metadata³ y otros, usados aún hoy día. Estos estándares pretendían sentar las bases para tener APIs que permitieran funcionalidades como la notificación al LMS indicando el comienzo, progreso y fin de los contenidos.

La **segunda generación** de plataformas (de 1999 en adelante) expandieron los logros de los sistemas predecesores y empezaron a corregir sus fallos. Ejemplos de esta generación son Moodle, Sakai y WebCT/Blackboard. En términos de evolución, empezaron a proveer un cambio hacia el diseño modular y a reconocer la necesidad de intercambios semánticos. Se empezó no sólo a compartir contenido sino también objetos, secuencias e información relacionada con el aprendizaje. Estándares y especificaciones empezaron a surgir durante esta época como SCORM (Shareable Content Object Reference Model)⁴, IMS Content Packaging e IMS Learning Design⁵, que permitían la exportación e importación de cursos o parte de cursos así como la interoperabilidad entre diferentes LMS. Dado que el diseño empezó a ser más modular, se hizo más sencillo integrar nueva funcionalidad en los LMS.

Y hoy día nos encontramos con la **tercera generación**. Continúa la evolución hacia la siguiente generación de plataformas LMS. El objetivo es aplicar *frameworks* de servicios al diseño modular ya implantado. Separar LMS de LCMS (ver diferencias en [112]) provee el soporte necesario para una gran interoperabilidad, en la cual los sistemas pueden, además de compartir contenido y escenarios de aprendizaje, intercambiar herramientas, funcionalidades, conceptos semánticos y controlar dinámicamente los cursos. Esto conlleva un mayor rango de tipo de información a manejar, como información de usuario, el contexto en el que se encuentra, secuencias de trabajo, etc. que pueden usarse para permitir a los usuarios construir plataformas personalizadas para sus necesidades específicas y de una manera dinámica. La idea por tanto es tener LMS que puedan “comunicarse” con otras plataformas ofreciendo una mayor variedad de servicios, permitiendo a los usuarios elegir los que ellos necesitan en función de sus requerimientos.

La siguiente generación de plataformas e-learning introducen por tanto la interoperabilidad de información y flexibilidad como puntos clave. La información no será sólo exportable o importable entre diferentes entornos, sino también entre entornos heterogéneos. Algunos sistemas permiten enviar información del usuario (estado del conocimiento, niveles de habilidad, preferencias, etc.) o infor-

¹<http://www.dublincore.org/>

²<http://www.imsglobal.org/specifications.html>

³<http://ltsc.ieee.org/wg12/>

⁴<http://www.adlnet.gov>

⁵<http://www.imsglobal.org/specifications.html>

mación de contenido (metadatos sobre el área de enseñanza, aspectos técnicos del contenido, etc.) entre diferentes entornos. Sin embargo, el siguiente paso es hacer que las plataformas puedan entender la información semántica, cómo procesarla y que ésta sea utilizada en diferentes entornos. Para llegar a este nivel de interoperabilidad, los entornos deben intercambiar tanto la información sintáctica como semántica.

Para finalizar esta evolución en los LMS, se ha incluido en la figura 4.1 de la página 19 un resumen gráfico de las tecnologías utilizadas en cada generación de LMS [20].

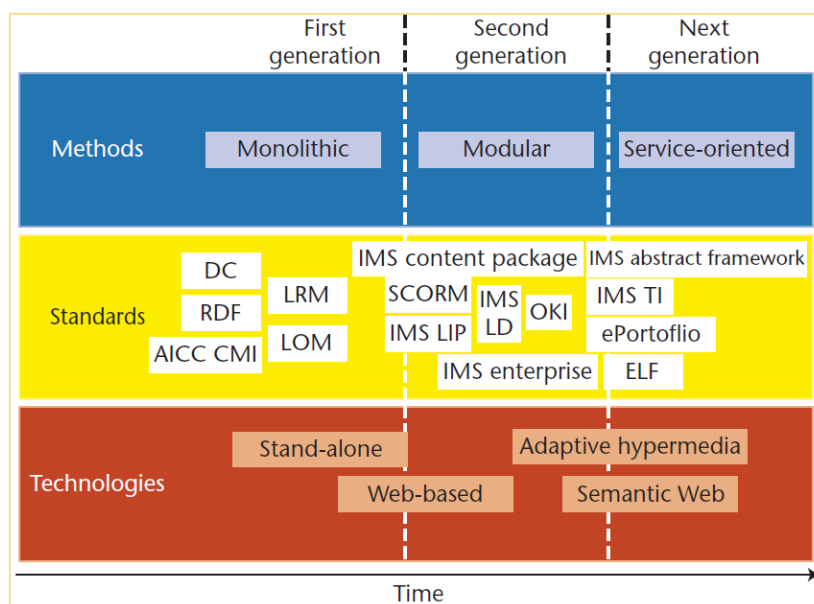


Figura 4.1: Generación de los LMS

Se invita al lector a acudir a la referencia ya citada [20] para profundizar en el avance de los LMS y hacia dónde se dirigen éstos.

Capítulo 5

Revisión Sistemática de la Literatura — Método de investigación

Como se ha comentado al principio del trabajo, en 1.3, el método de investigación utilizado en este estudio es una revisión sistemática de la literatura basada en las guías propuestas por [51] y [11].

A lo largo de este capítulo, se aplicarán las guías fase a fase, describiendo cada una de ellas y dando en los casos que corresponda una serie de resultados o productos que serán así mismo presentados.

5.1. Planear

En esta sección, nos centraremos en la fase de planificación que incluye la definición de los objetivos de la investigación y la forma en la que se llevará a cabo la revisión. En concreto, presentaremos en las siguientes subsecciones lo siguiente:

- Las cuestiones realizadas para la investigación.
- La fuente de datos junto con las cadenas de búsqueda.
- El criterio de inclusión y exclusión
- Aplicar un criterio de calidad.
- Los datos extraídos de cada estudio seleccionado.
- El proceso utilizado para dirigir la revisión sistemática de la literatura.

5.1.1. Cuestiones realizadas

El primer paso en la fase de planificación ha sido definir el objetivo principal de esta revisión sistemática de la literatura. Dicho objetivo es identificar y analizar el estado del arte de MDE (o MDA) relacionado con el aprendizaje basado en la tecnología o aprendizaje potenciado por la tecnología (TEL) y LMS.

Para alcanzar dicho objetivo, se han definido una serie de cuestiones o preguntas de investigación (PI):

- *PI1: ¿Qué es MDE, MDA, TEL y LMS?* Lo primero y más importante es saber qué es cada cosa y para qué sirve hoy día, además de averiguar sus aplicaciones actuales.
- *PI2: ¿Qué relación hay entre MDE (o MDA) y TEL?* Es necesario encontrar la relación entre ambos conceptos y así obtener información sobre cómo MDE (o MDA) pueden ayudar en el aprendizaje basado en la tecnología.
- *PI3: ¿Qué relación hay entre TEL y LMS?* Queremos saber cómo el concepto más amplio TEL tiene cabida en uno más específico, como es LMS.
- *PI4: ¿Qué aplicación tiene MDE (o MDA) en LMS?* Después de todo, lo que vamos buscando es la aplicación de las tecnologías en LMS.

5.1.2. Fuente de datos y cadenas de búsqueda

La fase de planificación también incluye el enumerar las fuentes de datos donde se han realizado las búsquedas para obtener estudios o trabajos previos [51]. Para este trabajo, se han aplicado las búsquedas en las siguientes librerías o espacios digitales, usando como idioma el inglés:

- *IEEEExplore* [IEEEX]: <http://ieeexplore.ieee.org/>
- *Science Direct* [SD]: <http://www.sciencedirect.com/>
- *SpringerLink* [SL]: <http://www.springerlink.com/>

Paralelamente a Science Direct, se han realizado búsquedas en Scopus¹, de Elsevier, y Google Scholar², de Google.

Aparte de estas páginas web, también se ha recurrido a buscadores como Google³, DuckDuckGo⁴, bing⁵ o la web Wikipedia⁶, cuando se ha necesitado algún artículo muy concreto o incluso cuando algún PDF no se encontraba en las librerías digitales.

¹<http://www.scopus.com>

²<http://scholar.google.com/>

³<http://www.google.com/>

⁴<http://www.duckduckgo.com/>

⁵<http://www.bing.com/>

⁶<http://en.wikipedia.org>

También se ha utilizado como fuente de datos la web de OMG⁷, o más concretamente el listado de especificaciones⁸ de dicho consorcio.

Finalmente, se ha utilizado documentación previa, aportada por profesores, que ha servido para la investigación llevada a cabo. Esta documentación trata temas relacionados con esta investigación y ha sido utilizada como punto de partida del trabajo.

Las cadenas de búsqueda realizadas aparecen listadas en la tabla 5.1 de la página 22. En la tabla, *TAK* indican *Title*, *Abstract*, *Keywords*, y *A* indica *All*. Ambos casos se refieren a los campos de búsqueda utilizados.

Cuadro 5.1: Fuentes de datos y cadenas de búsquedas realizadas

Libería	Búsqueda	Ámbito
IEEEEX	(mda OR mde) AND software	TAK
IEEEEX	(mda OR mde OR tel) AND software AND learning	TAK
IEEEEX	Technology-Enhanced Learning AND software	TAK
IEEEEX	(mda OR mde OR tel) AND systematic review	TAK
IEEEEX	lms AND software	TAK
IEEEEX	Technology-Enhanced Learning AND software	A
IEEEEX	(lms OR Learning Management System) AND tel	TAK
SD	(mda OR mde) AND software	TAK
SD	(mda OR mde OR tel) AND software AND learning	TAK
SD	Technology-Enhanced Learning AND software	TAK
SD	(mda OR mde OR tel) AND systematic review	TAK
SD	lms AND software	TAK
SD	(mda OR mde) AND software	TAK+A
SD	Technology-Enhanced Learning AND software AND (mda OR mde)	A
SD	(lms OR Learning Management System) AND tel	TAK
SL	(mda OR mde) AND software	TAK
SL	(mda OR mde OR tel) AND software AND learning	TAK
SL	Technology-Enhanced Learning AND software	TAK
SL	(mda OR mde OR tel) AND systematic review	TAK
SL	lms AND software	TAK
SL	(mda OR mde OR tel) AND software AND learning	A
SL	Technology-Enhanced Learning AND software AND (mda OR mde)	A
SL	(lms OR Learning Management System) AND tel	TAK

⁷<http://www.omg.org/>

⁸<http://www.omg.org/spec/index.htm>

5.1.3. Criterio de inclusión y exclusión

Las búsquedas realizadas han intentado abarcar todo el ámbito de esta investigación, pero es posible (y de hecho, ocurre) que el resultado contenga artículos, referencias, publicaciones, estudios o, en general, información que no nos sirva.

Siguiendo la guía de [51], es necesario definir criterios de inclusión y exclusión basado en las preguntas de investigación (ver 5.1.1) para poder filtrar la información.

Para este Trabajo de Investigación, se han incluido estudios publicados desde el año 2000 en adelante, ya que años anteriores pueden contener información caduca, y que cumplan al menos con los siguientes criterios:

- El *abstract* nos debe permitir concluir que el estudio trata sobre MDE, MDA, TEL o LMS, siendo su ámbito de aplicación la ingeniería software.
- Una lectura rápida de la introducción debe servir para concluir que el estudio explica algunos de los conceptos (MDE, MDA, TEL o LMS) y que la información va orientada al ámbito de la enseñanza o el aprendizaje.
- Las palabras claves incluyen uno o varios de los conceptos de nuestro interés: MDE, MDA, TEL o LMS.
- Los estudios deben estar redactados y su idioma deber ser el inglés.

Por otro lado, tras tener estos artículos incluidos, muchos de ellos, tras ser analizados, eran excluidos atendiendo a criterios como:

- Estudios demasiado ambiguos o genéricos que no permitan obtener información relevante.
- Estudios muy específicos que tratan una tecnología concreta, o un diseño o modelado particular, entendiendo esto último como el modelo que sale tras aplicar una ingeniería dirigida por modelos.
- Estudios que tratan los conceptos ya mencionados (MDE, MDA, TEL o LMS) sin estar enfocados al aprendizaje y la enseñanza.
- Estudios basados en otros estudios que no aportan nada nuevo.

5.1.4. Asegurar la calidad

Una vez que se ha obtenido un número de trabajos o estudios que satisfacen los criterios de inclusión y exclusión, deberíamos de ser capaces de asegurar o afirmar que cumplen con una mínima calidad y son válidos para este Trabajo de Investigación. Es básico el poder analizar qué estudios son mejores que otros en términos de calidad.

Siguiendo, como hasta ahora, la guía de [51], se han definido las siguientes preguntas o cuestiones que permiten asegurar dicha calidad.

- ¿Cómo de claro y coherente es el trabajo o estudio?
- ¿Cómo de claro es el objetivo que busca el trabajo o estudio?
- ¿Cómo de clara es la conclusión del trabajo o estudio?
- ¿Cómo de bueno es el trabajo en comparación con otros similares?

Así mismo, la respuesta a cada pregunta se ha evaluado de 1 a 3, siendo 1 un estudio o información poco relevante o de baja calidad, y 3 un estudio o información muy relevante o de alta calidad.

5.1.5. Extracción de datos y análisis

Los criterios de inclusión y exclusión (ver 5.1.3), junto a las preguntas orientadas a asegurar la calidad (ver 5.1.4), nos han servido para obtener todos los artículos y estudios que se analizan en este Trabajo de Investigación.

De cada artículo, se ha recogido título y autores, *abstract*, publicación (cuando ha sido posible) y año de publicación (cuando ha sido posible). $\text{\LaTeX 2}_{\epsilon}$, con su gestor de bibliografía BibTEX, y Mendeley junto a JabRef, con sus herramientas de extracción y exportación de información, han sido de mucha ayuda para esta tarea.

Además, con el objetivo de llegar a un análisis más profundo de los estudios y artículos que se han recabado, se ha decidido extraer una determinada información de cada estudio primario, que son aquellos que cumplen el criterio de inclusión y no han sido excluidos por el criterio de exclusión, así como aquellos que han pasado un mínimo control de calidad (ver 5.1.6).

La información que se ha decidido extraer de cada estudio primario es la siguiente:

- P1 Tipo de concepto que se está tratando. Es decir, si el artículo se centra en MDA, MDE, TEL, LMS o DSL. Se indicará, si fuera necesario, algún otro concepto novedoso que sea tratado por el artículo o estudio.
- P2 Qué arquitectura, si existiera, es la que se ha utilizado o se ha tratado.
- P3 Qué tecnología o plataforma software, si existiera, es la que se ha utilizado o se ha tratado.
- P4 Metodología o ingeniería que abarca el estudio o artículo.

Cabe destacar que, como se ha dicho anteriormente, ha sido necesario en muchos casos leer en mayor o menor detalle los resultados obtenidos para descartar o no la información, y quedarnos únicamente con lo relevante.

5.1.6. Proceso para dirigir la revisión

El último paso de la fase de planificación es definir el proceso para conducir o dirigir la revisión.

Para esta revisión sistemática de la literatura, se ha utilizado el proceso descrito en [80] . Básicamente consiste en tres fases, como puede verse en la figura 5.1 de la página 25.

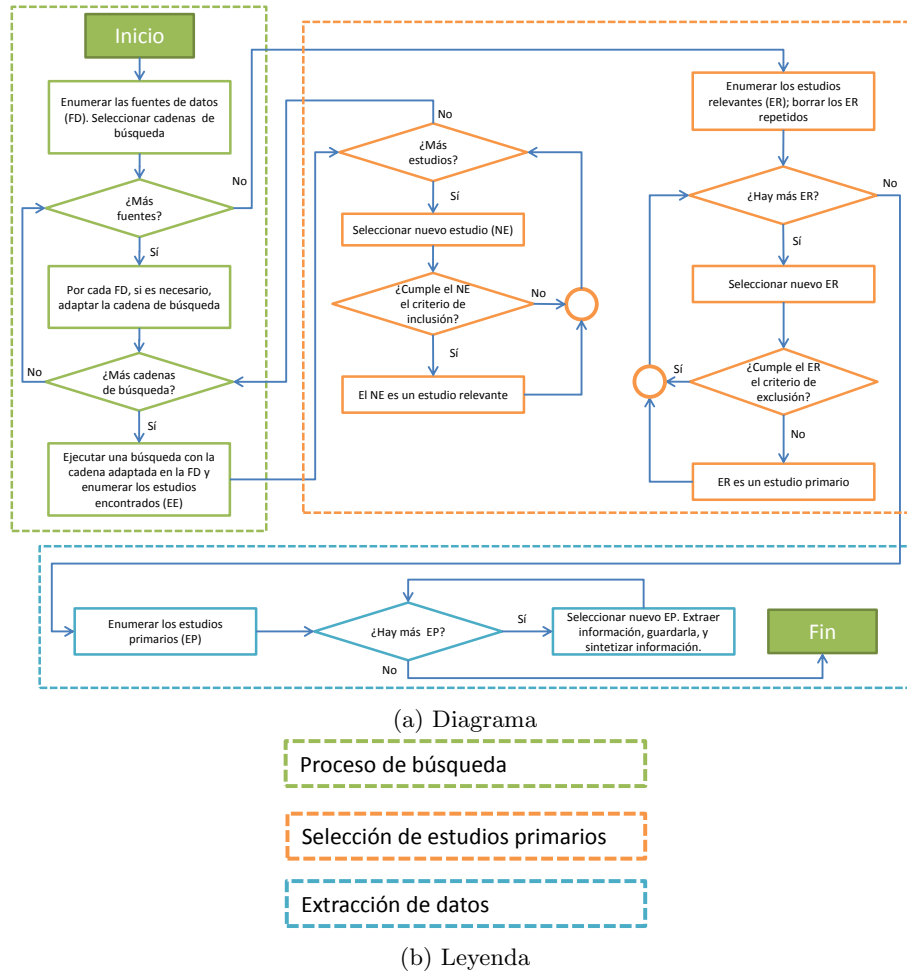


Figura 5.1: Proceso para dirigir la revisión

Usando este proceso, tenemos tres fases: proceso de búsqueda (recuadro punteado verde), selección de estudios primarios o principales (recuadro punteado naranja), y extracción de datos (recuadro punteado azul).

Proceso de búsqueda El primer paso en el proceso de búsqueda consiste en enumerar las fuentes de datos (FD) así como seleccionar las cadenas de búsqueda que se usarán en la búsqueda de estudios o información. Una vez que las fuentes de datos han sido identificadas, hay que adaptar a cada motor de búsqueda (si es necesario) las consultas a realizar. A continuación, se ejecutan las cadenas de búsquedas para la búsqueda de estudios o información en las correspondientes FD. Como se mencionó anteriormente, para este estudio se han definido una serie de cadenas de búsqueda (ver 5.1.2), así como listado las fuentes de datos (ver tabla 5.1 de la página 22) e indicado el ámbito o campos de búsqueda de aplicación.

El siguiente paso consiste en el almacenamiento y enumeración de todos los estudios encontrados (EE) o de la información obtenida por cada FD. Como se ha mencionado en la sección anterior (ver 5.1.5), este paso implica la siguiente extracción de datos para cada estudio: título y autores, *abstract*, publicación (cuando ha sido posible) y año de publicación (cuando ha sido posible).

Selección de estudios primarios A continuación, se aborda el proceso de selección de estudios primarios o principales. Esto implica comprobar si cada nuevo estudio (NE) obtenido cumple el criterio de inclusión definido en la sección 5.1.3. Es decir, artículos del año 2000 en adelante y que cumplan lo siguiente:

- El *abstract* nos debe permitir concluir que el estudio trata sobre MDE, MDA, TEL o LMS, siendo su ámbito de aplicación la ingeniería software.
- Una lectura rápida de la introducción debe servir para concluir que el estudio explica algunos de los conceptos (MDE, MDA, TEL o LMS) y que la información va orientada al ámbito de la enseñanza o el aprendizaje.
- Las palabras claves incluyen uno o varios de los conceptos de nuestro interés: MDE, MDA, TEL o LMS.
- Los estudios deben estar redactados y su idioma deber ser el inglés.

Si el estudio o información analizada cumple con el criterio de inclusión, **se convierte en un estudio relevante para esta revisión sistemática de la literatura.**

Después de haber identificado los estudios relevantes (RE) obtenidos de las diferentes fuentes y librerías digitales, es el momento de abordar la segunda parte del proceso de selección de estudios primarios. Esta segunda parte consiste en dos tareas:

- Eliminar la posible información redundante o duplicada que aparece a raíz de haber ejecutado la misma búsqueda en diferentes motores de búsqueda.
- Cada estudio no duplicado se evalúa de acuerdo al criterio de exclusión definido en la sección 5.1.3 para identificar aquellos estudios que deben ser excluidos del conjunto final de estudios primarios (EP) que son utilizados para la fase final de todo el proceso: extracción de datos.

Extracción de datos En esta última fase final, cada estudio primario (EP) se analiza en detalle para extraer la información definida en la sección 5.1.5. Según el estudio (artículo, referencia, publicación, etc.), se evalúa la necesidad de analizar e investigar las referencias bibliográficas citadas por cada EP, con la idea de buscar otros trabajos que pueden ser relevantes para este trabajo de investigación.

5.2. Ejecutar

Tras la fase anterior, descrita ampliamente en la sección 5.1, es el momento de ejecutar dicha planificación.

Esta sección presenta los resultados principales obtenidos tras la revisión sistemática de la literatura realizada de acuerdo al método ya descrito.

5.2.1. Búsquedas de estudios

Como se ha mencionado anteriormente cuando se ha descrito el proceso seguido para conducir la revisión (ver 5.1.6), el primer paso ha sido ejecutar las búsquedas en cada fuente de datos con la correspondiente cadena adaptada a la sintaxis del motor de búsqueda.

La tabla 5.1 de la página 22 recoge las cadenas de búsqueda así como el ámbito o aplicación de dichas búsquedas. Ejecutando esto, se obtiene lo que se ha recogido en las tablas 5.2 y 5.3 de las páginas 27 y 28, respectivamente.

La tabla 5.2 contiene las librerías digitales, las cadenas ejecutadas, y el número de resultados obtenidos para cada cadena de búsqueda. Por otro lado, la tabla 5.3 es la suma total de resultados por cada librería digital.

Como puede observarse, se han recogido un total de 3448 estudios. Según el caso, y si se ha visto necesario, se han usado los ordenamientos y filtrados que ofrecen los motores de búsqueda para quedarnos sólo con lo más reciente o lo más citado. Esto ha sido necesario cuando los resultados obtenidos eran demasiado numerosos (más de 500 resultados). Filtrar u ordenar facilita el quedarse con los estudios más importantes.

El siguiente paso, siguiendo el proceso de esta revisión, y bocetado en la figura 5.1 de la página 25, ha sido la selección de estudios relevantes siempre y cuando cumplan los criterios de inclusión (ver 5.1.3). Aplicando dichos criterios, de los 3448 estudios, nos hemos quedado con 74 artículos o estudios, o lo que es lo mismo, un 2,15 % del total.

Cuadro 5.2: Resultados de las búsquedas obtenidos por cada librería y cadena de búsqueda

Librería	Búsqueda	Resultados
IEEEEX	(mda OR mde) AND software	656
IEEEEX	(mda OR mde OR tel) AND software AND learning	49
continúa...		

Cuadro 5.2 – continuación...

Libería	Búsqueda	Resultados
IEEEEX	Technology-Enhanced Learning AND software	76
IEEEEX	(mda OR mde OR tel) AND systematic review	1
IEEEEX	lms AND software	330
IEEEEX	Technology-Enhanced Learning AND software	910
IEEEEX	(lms OR Learning Management System) AND tel	23
SD	(mda OR mde) AND software	95
SD	(mda OR mde OR tel) AND software AND learning	6
SD	Technology-Enhanced Learning AND software	21
SD	(mda OR mde OR tel) AND systematic review	22
SD	lms AND software	23
SD	(mda OR mde) AND software	501
SD	Technology-Enhanced Learning AND software AND (mda OR mde)	365
SD	(lms OR Learning Management System) AND tel	2
SL	(mda OR mde) AND software	277
SL	(mda OR mde OR tel) AND software AND learning	8
SL	Technology-Enhanced Learning AND software	28
SL	(mda OR mde OR tel) AND systematic review	1
SL	lms AND software	33
SL	Technology-Enhanced Learning AND software AND (mda OR mde)	17
SL	(lms OR Learning Management System) AND tel	4

Cuadro 5.3: Resultados totales de las búsquedas

Libería	Num. resultados	Estudios relevantes	% ER	% Total
IEEEEX	2045	32	1,56 %	43 %
SD	1035	22	2,13 %	29 %
SL	368	20	5,43 %	27 %
Total	3448	74	2,15 %	100 %

Parte de la fase de selección de estudios primarios es eliminar la posible información redundante o duplicada que aparece a raíz de haber ejecutado la misma búsqueda en diferentes motores de búsqueda. Se ha sido cuidadoso a la

hora de almacenar la información que se iba obteniendo de las librerías digitales, con lo cual, en este Trabajo de Investigación, de todos los estudios recogidos, no hay ninguno repetido. Sí que es cierto que han aparecido durante el proceso. Es decir, al buscar, por ejemplo, en IEEEExplore y en Science Direct, en ocasiones, el mismo estudio ha aparecido en los dos motores de búsqueda, pero directamente se ha cogido de un único sitio.

5.2.2. Selección y listado de estudios primarios

Llegados a este punto, y tras tener los artículos o estudios identificados y sin repetición, éstos han sido analizados y excluidos según los criterios definidos en la sección 5.1.3. Este paso nos da finalmente los estudios primarios listados en las tablas correspondientes a IEEEExplore 5.4 (página 29), Science Direct 5.5 (página 30) y SpringerLink 5.6 (página 31).

Cuadro 5.4: Listado de estudios primarios de IEEEExplore

ID	Título	Año
1	ACM-learning architecture: A normative proposal for technology enhanced education for developing countries [98]	2012
2	Adapting LMS Architecture to the SOA: An Architectural Approach [37]	2009
3	Adapting Software Development Process towards the Model Driven Architecture [75]	2008
4	Adaptive and Intelligent Systems for Collaborative Learning Support: A Review of the Field [59]	2011
5	An architecture-oriented model-driven requirements engineering approach [55]	2011
6	An MDA-Based System Development Lifecycle [6]	2008
7	CMS-Based Web-Application Development Using Model-Driven Languages [92]	2009
8	Comparing open-source e-learning platforms from adaptivity point of view [87]	2009
9	Contextualizing learning scenarios according to different Learning Management Systems [27]	2011
10	Delivering IMS Learning Design Activities via Mobile Devices [89]	2007
11	Developing applications using model-driven design environments [9]	2006
12	Empirical assessment of MDE in industry [43]	2011
13	Enhancing IMS LD Units of Learning Comprehension [62]	2009
14	IMS learning design support for the formalization of collaborative learning patterns [54]	2004
15	Investigation of Virtual Learning Environment in the Context of Web 2.0 [84]	2010
continúa...		

Cuadro 5.4 – continuación...

ID	Título	Año
16	MDA-based development of e-learning system [110]	2003
17	MDE between Promises and Challenges [35]	2009
18	Model driven process engineering [13]	2001
19	Model-Driven approach to Software Architecture design [79]	2009
20	Model-driven Development of Complex Software : A Research Roadmap Model-driven Development of Complex Software : A Research Roadmap [32]	2007
21	Model-driven development of learning objects [100]	2011
22	Model-Driven Evolution of Software Architectures [39]	2007
23	Models and Transformations in MDA [101]	2009
24	Open Access to Science Education Resources and Learning Designs in Europe [90]	2012
25	Open source learning management systems in e-learning and Moodle [8]	2010
26	Process patterns for MDA-based software development [5]	2010
27	Software Construction of an Authoring Tool for Adaptive E-learning Platforms [108]	2009
28	Strategies for Improving MDA-Based Development Processes [36]	2010
29	The Impact of the Model-Driven Approach to Software Engineering on Software Engineering Education [41]	2009
30	The pragmatics of model-driven development [97]	2003
31	Tools for MDA Software Development: Evaluation Criteria and Set of Desirable Features [16]	2008
32	Towards outcome based learning: An engineering education case [74]	2011

Cuadro 5.5: Listado de estudios primarios de Science Direct

ID	Título	Año
33	A Domain-Specific Modeling approach for supporting the specification of Visual Instructional Design Languages and the building of dedicated editors [53]	2010
34	A framework for model-driven development of information systems: Technical decisions and lessons learned [107]	2012
35	A survey on search-based software design [85]	2010
36	A systematic review of UML model consistency management [56]	2009
37	An adaptive and personalized open source e-learning platform [106]	2010
continúa...		

Cuadro 5.5 – continuación...

ID	Título	Año
38	Barbed Model-Driven Software Development: A Case Study [71]	2008
39	CIAT-GUI: A MDE-compliant environment for developing Graphical User Interfaces of information systems [70]	2012
40	Context and deep learning design [12]	2012
41	Definition and use of Computation Independent Models in an MDA-based groupware development process [33]	2007
42	Functional metamodels for systems and software [103]	2009
43	High-level brokerage services for the e-learning domain [4]	2003
44	Integration of web based environment for learning discrete simulation in e-learning system [21]	2012
45	Language engineering techniques for the development of e-learning applications [63]	2009
46	Language support for model-driven software development [17]	2008
47	Model-driven architecture for embedded software: A synopsis and an example [48]	2008
48	Model-driven engineering techniques for the development of multi-agent systems [34]	2012
49	Model-driven performance analysis of rule-based domain specific visual models [105]	2013
50	Modelling software development methodologies: A conceptual foundation [38]	2007
51	Semantic model-driven architecting of service-based software systems [78]	2007
52	Supporting software evolution in component-based FOSS systems [24]	2011
53	Taking context into account in conceptual models using a Model Driven Engineering approach [14]	2011
54	Towards a Megamodel to Model Software Evolution Through Transformations [29]	2005

Cuadro 5.6: Listado de estudios primarios de SpringerLink

ID	Título	Año
55	A Computer Science Perspective on Technology-Enhanced Learning Research [102]	2009
56	A model-driven architectural framework for integration-capable enterprise application product lines [52]	2006
57	A model-driven architecture approach for developing e-learning platform [18]	2010
58	A Proposal for Goal Modelling Using a UML Profile [40]	2008
continúa...		

Cuadro 5.6 – continuación...

ID	Título	Año
59	An Architecture for e-Learning System with Computational Intelligence [2]	2007
60	Building Quality into Learning Management Systems – An Architecture – Centric Approach [7]	2003
61	Competence models in technology-enhanced competence-based learning [88]	2008
62	Computational Intelligence Infrastructure in Support for Complex e-Learning Systems [15]	2010
63	Developing BP-driven web applications through the use of MDE techniques [104]	2010
64	Enterprise Portal Technology in Computer-Supported Collaborative Learning [46]	2012
65	Formalizing MDA components [30]	2006
66	GLUE!-PS: A multi-language architecture and data model to deploy TEL designs to multiple learning environments [82]	2011
67	Graph transformations in OMG's model-driven architecture [47]	2004
68	Learning Assessment Using Wikis: Integrated or LMS Independent? [31]	2010
69	Learning object repurposing for various multimedia platforms [69]	2011
70	Model Driven Engineering [50]	2002
71	Software and Standards in an Emerging Domain [83]	2010
72	Technology-Enhanced Support for Learning Conceptual Modeling [96]	2012
73	The digital course training workshop for online learning and teaching [49]	2012
74	Using Open Source Software for Improving Dialog in Computer Science Education – Case Mozambique University [64]	2011

5.2.3. Asegurar la calidad

El último paso antes de pasar a analizar y extraer la información de los estudios es poner en juego las cuestiones que se propusieron en el punto 5.1.4:

- ¿Cómo de claro y coherente es el trabajo o estudio? (cuestión #1)
- ¿Cómo de claro es el objetivo que busca el trabajo o estudio? (cuestión #2)
- ¿Cómo de clara es la conclusión del trabajo o estudio? (cuestión #3)

- ¿Cómo de bueno es el trabajo en comparación con otros similares? (cuestión #4)

El objetivo es ser capaces de asegurar o afirmar que los estudios primarios cumplen con una mínima calidad y son válidos para este Trabajo de Investigación. Los estudios han sido por tanto analizados en base a las cuestiones anteriores. Cada pregunta se ha puntuado de 1 a 3. La tabla 5.7 (página 33) recoge las puntuaciones dadas a cada estudio así como su puntuación total.

Cuadro 5.7: Puntuación de calidad para los estudios primarios

Identificador	C#1	C#2	C#3	C#4	Total
1	2	1	1	1	1,25
2	2	2	2	2	2,00
3	3	2	2	2	2,25
4	2	2	1	1	1,50
5	3	2	2	2	2,25
6	3	2	3	2	2,50
7	3	2	2	2	2,25
8	2	3	2	1	2,00
9	3	3	2	3	2,75
10	1	1	1	1	1,00
11	2	2	1	2	1,75
12	2	1	2	1	1,50
13	2	2	2	2	2,00
14	1	1	2	1	1,25
15	1	1	1	1	1,00
16	3	3	3	2	2,75
17	3	3	3	2	2,75
18	2	2	2	1	1,75
19	2	1	2	1	1,50
20	3	3	3	2	2,75
21	1	1	1	1	1,00
22	3	3	2	1	2,25
23	2	1	1	2	1,50
24	1	1	1	1	1,00
25	2	1	1	1	1,25
26	2	2	2	1	1,75
27	2	2	1	1	1,50
28	2	2	2	1	1,75
29	2	2	2	1	1,75
30	2	2	2	2	2,00
31	2	2	2	2	2,00
32	2	2	2	1	1,75
33	1	1	1	1	1,00
continúa...					

Cuadro 5.7 – continuación...

Identificador	C#1	C#2	C#3	C#4	Total
34	2	1	2	2	1,75
35	3	2	3	3	2,75
36	1	1	2	2	1,50
37	2	3	1	1	1,75
38	1	1	1	1	1,00
39	2	1	2	2	1,75
40	2	1	1	2	1,50
41	1	1	1	1	1,00
42	2	2	1	2	1,75
43	2	1	1	2	1,50
44	1	1	1	2	1,25
45	2	2	2	1	1,75
46	2	1	1	1	1,25
47	2	2	2	1	1,75
48	1	1	1	2	1,25
49	1	1	1	1	1,00
50	2	1	1	2	1,50
51	2	1	1	1	1,25
52	1	1	1	1	1,00
53	1	1	1	2	1,25
54	2	2	1	2	1,75
55	3	2	1	1	1,75
56	1	1	1	1	1,00
57	2	2	2	2	2,00
58	2	2	2	1	1,75
59	2	3	2	1	2,00
60	2	2	1	1	1,50
61	1	1	1	1	1,00
62	1	1	1	1	1,00
63	2	2	2	2	2,00
64	2	1	1	1	1,25
65	1	1	1	2	1,25
66	2	2	1	2	1,75
67	2	2	2	2	2,00
68	2	2	2	1	1,75
69	2	2	2	1	1,75
70	2	2	2	1	1,75
71	2	1	1	1	1,25
72	1	1	1	1	1,00
73	1	1	1	1	1,00
74	2	2	2	1	1,75

Este Trabajo de Investigación se ha centrado en aquellos estudios o artículos que han tenido una **puntuación de calidad de 1.75 o más**. Los demás estudios, a pesar de haberse tenido en cuenta y haber sido analizados, se han dejado en un segundo plano. La extracción de información y análisis se ha centrado por tanto en esos estudios o artículos que cumplen con un mínimo de calidad.

En todo caso, los estudios que han quedado por debajo de 1.75 tras ser evaluados con los criterios de calidad ya expuestos son igualmente útiles en cuanto a la revisión sistemática de la literatura. Quedan por tanto recogidos y listados en este Trabajo de Investigación, y son totalmente válidos para conocer y ahondar más en el estado del arte que es objeto de este trabajo.

5.2.4. Otros artículos y estudios

La tabla 5.7 de la página 33 contiene los estudios primarios que han sido considerado válidos (en cuestiones de calidad y contenido) para este Trabajo de Investigación.

No obstante, hay otros artículos que, como se comentó en la sección 5.1.2, fueron facilitados por los profesores para iniciar este trabajo o bien fueron encontrados en internet sin ningún patrón de búsqueda en concreto. Estos estudios son listados en la tabla 5.8 de la página 35 y han sido también utilizados en esta revisión sistemática de la literatura.

Cuadro 5.8: Otros estudios recogidos por esta revisión sistemática de la literatura

ID	Título	Año
75	A Domain-Specific Language for Web APIs and Services Mashups [65]	2007
76	A Framework for the Conceptualization of Approaches to “Create-by-Reuse” of Learning Design Solutions [42]	2007
77	A Grid Based Open Framework for Technology Enhanced Learning [3]	2008
78	An extensible approach to visually editing adaptive learning activities and designs based on services [25]	2010
79	An MDA approach for the development of data warehouses [66]	2008
80	An MDA approach to knowledge engineering [81]	2012
81	Application of model-driven engineering (MDA) for the construction of a tool for domain-specific modeling (DSM) and the creation of modules in learning management systems (LMS) platform independent [60]	2011
82	Authoring and reengineering of IMS Learning Design units of learning [61]	2009
continúa...		

Cuadro 5.8 – continuación...

ID	Título	Año
83	Building applications with domain-specific markup languages: a systematic approach to the development of XML-based software [99]	2003
84	Design approaches in technology-enhanced learning [73]	2007
85	Domain specific language for the generation of learning management systems modules [72]	2012
86	Domain-specific languages: an annotated bibliography [22]	2000
87	Extending IMS-LD Capabilities: A Review, a Proposed Framework and Implementation Cases [58]	2012
88	FlexoLD: Un Lenguaje Específico de Dominio para Diseños de Aprendizaje [26]	2010
89	In search of a basic principle for model driven engineering [10]	2004
90	MDA-based development of e-learning system [110]	2003
91	Model Driven Engineering [94]	2006
92	Model-Driven Engineering as a new landscape for traceability management: A systematic literature review [91]	2012
93	Model-Driven Open Source Software Development – The Open Models Approach [1]	2009
94	Modelling Knowledge and Game Based Learning: Model Driven Approach [69]	2011
95	Service-Oriented E-Learning Platforms. From Monolithic Systems to Flexible Services [20]	2007
96	Survey of Educational Modelling Languages (EMLs) [86]	2002
97	Teacher tool for visualization and management of a technology-enhanced learning environment [19]	2011
98	Technology Enhanced Informal Science Learning and Engagements: A Typology of Activity [93]	2012
99	The educational modeling languages in Instructional Design: Towards a UML applications [44]	2011
100	The Pragmatics of Current E-Learning Standards [23]	2007
101	Using Learning Objects in Games [68]	2010
102	When and how to develop domain-specific languages [67]	2005
103	Why university lecturers enhance their teaching through the use of technology: a systematic review [45]	2011

5.3. Analizar y Extraer datos

Tras la ejecución la planificación según lo descrito en la sección 5.2, se han obtenido los estudios listados en las subsecciones 5.2.3 y 5.2.4.

Es por tanto el momento de analizar y extraer información y datos de cada uno de ellos. Tal y como se detalló en la sección 5.1.5, se ha decidido extraer lo

siguiente:

P1 Tipo de concepto que se está tratando. Es decir, si el artículo se centra en MDA, MDE, TEL, LMS o DSL. Se indicará, si fuera necesario, algún otro concepto novedoso que sea tratado por el artículo o estudio.

P2 Qué arquitectura, si existiera, es la que se ha utilizado o se ha tratado.

P3 Qué tecnología o plataforma software, si existiera, es la que se ha utilizado o se ha tratado.

P4 Metodología o ingeniería que abarca el estudio o artículo.

Los datos extraídos se han recogido en las tablas 5.9 y 5.10 de la páginas 37 y 39.

Cuadro 5.9: Análisis y Extracción de datos – parte 1

ID	P1	P2
2	LMS	OKI (Open Knowledge Initiative)
3	MDA	N/A
5	MDA	N/A
6	MDA	N/A
7	MDE	N/A
8	LMS	N/A
9	MDA; TEL; LMS	N/A
11	MDA; DSML	N/A
13	LMS	N/A
16	MDA	N/A
17	MDE; MDA	N/A
18	MDA	N/A
20	MDE	N/A
22	MDE	N/A
26	MDA	N/A
28	MDA	N/A
29	MDA; TEL	N/A
30	MDD	N/A
31	MDA	N/A
32	LMS	N/A
34	MDE	N/A
35	<i>Nota: artículo que trata muy globalmente el diseño software</i>	N/A
37	LMS	N/A
39	MDE; MDA; GUI	N/A
42	MDE	N/A
continúa...		

Cuadro 5.9 – continuación...

ID	P1	P2
45	<i>Nota: artículo que trata las aplicaciones e-learning</i>	N/A
47	MDA	N/A
54	MDE; MDA	N/A
55	TEL	N/A
57	MDA	N/A
58	MDE; MDA	N/A
59	LMS	N/A
63	MDE	N/A
66	TEL	GLUE!-PS (Group Learning Unified Environment – Pedagogical Scripting)
67	MDA	N/A
68	LMS	N/A
69	MDA; LMS	N/A
70	MDE; MDA	N/A
74	LMS	N/A
75	DSL	N/A
76	LMS; EML	N/A
77	TEL	N/A
78	LMS; EML; DSL	N/A
79	MDA	DW (Data warehouses)
80	MDA	N/A
81	MDE; LMS; DSL	N/A
82	LMS; EML	N/A
83	DSL; DSML	N/A
84	TEL	N/A
85	MDA; LMS; DSL	N/A
86	DSL	N/A
87	N/A	N/A
88	DSL	N/A
89	MDA	N/A
90	MDA	N/A
91	MDE	N/A
92	MDE	N/A
93	MDA; MDD	N/A
94	MDA; LMS; EGLO (Educational Game Learning Object)	N/A
95	<i>Nota: artículo que trata la evolución en LMS</i>	N/A
96	<i>Nota: artículo que trata EML</i>	N/A
97	N/A	N/A
98	TEL	N/A
continúa...		

Cuadro 5.9 – continuación...

ID	P1	P2
99	EML	N/A
100	<i>Nota: artículo que trata el e-learning en general</i>	N/A
101	MDA; LMS; EGLO (Educational Game Learning Object)	N/A
102	DSL	N/A
103	<i>Nota: revisión de la literatura sobre la tecnología y la enseñanza</i>	N/A

Cuadro 5.10: Análisis y Extracción de datos – parte 2

ID	P3	P4
2	Moodle	SOA; IMS LTI
3	N/A	MDA; SPEM; MSF; RUP
5	N/A	Una extensión de OpenUP
6	N/A	MDA-SDLC (MDA-based System Development Lifecycle)
7	N/A	CMS-IL (CMS Intermediate Language); CMS-ML (CMS Modeling Language)
8	Docebo; Moodle; Dokeos; Claroline; ATutor; ILIAS	SCORM
9	Gen-COM	SCORM; IMS-LD
11	GME (Generic Modeling Environment)	N/A
13	e-LD	EML; IMS-LD
16	J2EE	LTSA; SCORM
17	N/A	N/A
18	N/A	MOF (Meta-Object Facility)
20	N/A	N/A
22	N/A	MOF (Meta-Object Facility); SAAM (Software Architecture Assessment Method)
26	N/A	Patrón de proceso propuesto basado en MDA
28	N/A	N/A
29	N/A	MOF (Meta-Object Facility); UML
continúa...		

Cuadro 5.10 – continuación...

ID	P3	P4
30	N/A	N/A
31	N/A	N/A
32	Moodle	LOD (Learning Outcome Definitions); MLO (Meta-data for Learning Opportunities); PALO (Personal Achieved Learning Outcomes)
34	EMF	M2DAT; M2DAT-DB
35	N/A	N/A
37	N/A	AES (Adaptative E-Learning Systems)
39	GMF; EMF	MBUID (Model-Based User Interface Development)
42	N/A	MOF (Meta-Object Facility), OCL (Object Constraint Language)
45	e-LD	IMS-LD
47	N/A	ESML (Embedded System Modeling Language)
54	N/A	N/A
55	N/A	N/A
57	N/A	N/A
58	N/A	UML
59	N/A	Proactive LMS
63	Eclipse; BPMN2BPEL; ATL; MOFScript; EMF	Ingeniería Web
66	Moodle; LAMS; wikis	IMS-LD
67	GT (Graph Transformation)	MIC (Model-Integrated Computing)
68	Moodle; wikis	N/A
69	N/A	N/A
70	N/A	N/A
74	N/A	N/A
75	DSL propio: Swashup DSL	N/A
76	N/A	IMS-LD
77	N/A	N/A
78	FLEXO LD	IMS-LD
79	N/A	UML
80	N/A	CommonKADS
81	Moodle	MOF (Meta-Object Facility)
82	e-LD	UML
continúa...		

Cuadro 5.10 – continuación...

ID	P3	P4
83	N/A	ADDS (Approach to Document-based Development of Software)
84	N/A	N/A
85	Moodle; Claroline; Atutor; GMF	N/A
86	N/A	N/A
87	N/A	IMS-LD
88	FlexoLD	N/A
89	N/A	N/A
90	J2EE	N/A
91	N/A	N/A
92	N/A	N/A
93	N/A	OSS (Open Source Software); UML
94	N/A	MOF (Meta-Object Facility)
95	N/A	N/A
96	N/A	N/A
97	N/A	N/A
98	N/A	N/A
99	N/A	UML
100	N/A	UML
101	N/A	N/A
102	N/A	N/A
103	N/A	N/A

Los campos puestos como “N/A” indican que para ese artículo, no se puede extraer ese determinado dato. Además, hay que hacer notar que los artículos que provienen de la tabla 5.8 de la página 35 suelen ser más genéricos, dado que no fueron obtenidos mediante la revisión sistemática de la literatura. De ahí que se encuentren en ellos la mayoría de los campos “N/A”.

5.4. Discusión

En esta sección se da respuesta a las cuestiones planteadas en la sección 5.1.1 tras haber completado la revisión sistemática de la literatura.

5.4.1. PI1: ¿Qué es MDE, MDA, TEL y LMS?

Lo primero y más importante es saber qué es cada cosa y para qué sirve hoy día, además de averiguar sus aplicaciones actuales y futuras. Dado que este Trabajo de Investigación no sólo es una revisión sistemática de la literatura,

para responder a esta pregunta se ha optado por ir abordando cada concepto, uno a uno, haciendo referencia a los artículos y estudios de la revisión cuando ha sido necesario.

Por tanto, la respuesta, que no es fácilmente resumible en pocas palabras, puede encontrarse en los diferentes capítulos de este Trabajo de Investigación:

- Se empezó tratando MDA en la sección 2.1 de la página 4.
- Se explicó MDE y Eclipse en la sección 2.2 de la página 6.
- Se han tratado los DSL en la sección 2.3 de la página 9.
- Se trató TEL en el capítulo 3 de la página 12.
- Finalmente, los LMS quedan explicados en el capítulo 4 de la página 16.

5.4.2. PI2: ¿Qué relación hay entre MDE (o MDA) y TEL?

Es necesario encontrar la relación entre ambos conceptos y así obtener información sobre cómo MDE (o MDA) pueden ayudar en el aprendizaje basado en la tecnología.

La evolución de las tecnologías de la información y la comunicación, sumado al incremento de su utilización en la educación, ha permitido una expansión de las actividades relacionadas con el aprendizaje y un crecimiento de la importancia del aprendizaje basado en la tecnología (TEL) [27].

Durante la revisión sistemática de la literatura, se ha observado que es necesario garantizar la interoperabilidad de los sistemas TEL a lo largo de diferentes LMS para reducir esfuerzos de desarrollo, tiempo y costes. La necesidad de dicha interoperabilidad ha permitido la creación de diferentes estándares de tecnologías para el aprendizaje, como SCORM e IMS-LD. Son, de hecho, dos de los estándares más mencionados en la literatura [27].

Sin embargo, estos estándares tienen límites que conciernen a la personalización y expresividad del contexto donde se desarrolla TEL. Adoptar un determinado estándar también restringe al diseñador de elegir un LMS en particular. Y es precisamente aquí donde tiene su “hueco” MDA, que emerge como un *framework* para la ingeniería software para abordar el problema de la interoperabilidad de los sistemas cuando estos deben ser ejecutados en diferentes plataformas. MDA consigue esto separando lo técnico de lo puramente empresarial y propone técnicas para integrar ambos elementos. Además, como ya se ha explicado previamente, MDA permite la generación de código a partir de modelos software [27]. Y cuando aquí se habla de MDA, tal y como ya se ha comentado en el documento, podemos aplicar la misma idea a MDE.

Siguiendo las prácticas aconsejadas por MDA, un sistema TEL se diseña en un proceso de tres pasos. Primero, crear un modelo del sistema ideado usando un metamodelo específico. Este metamodelo permite una descripción precisa de las necesidades. Segundo, un motor de transformación del modelo con reglas bien

delimitadas para transformar el modelo en un LMS concreto. Tercero, el modelo específico puede ser desplegado en un LMS usando un generador automático.

Este tipo de procesos conlleva un número de tareas técnicas que no todos los diseñadores pueden realizar de forma sencilla o eficiente. Es por ello que hay que recurrir a herramientas que asistan dichos procesos. Éstas suelen conocerse como herramientas AID (*Automated Instructional Design*, en inglés), que guían a los diseñadores noveles en el proceso de creación de modelos y sus transformaciones. Un buen ejemplo puede encontrarse en [27].

Se puede por tanto afirmar, tras la revisión de la literatura, que MDE (o MDA) es aplicable en el contexto de diseño de cursos para LMS y que reduce significativamente el tiempo y esfuerzo en la construcción y despliegue de cursos sobre plataformas LMS [60].

5.4.3. PI3: ¿Qué relación hay entre TEL y LMS?

Queremos saber cómo el concepto más amplio TEL tiene cabida en uno más específico, como es LMS.

LMS es, en cierto modo, la parte tecnológica de TEL. Son conceptos íntimamente relacionados. Según la literatura estudiada, la tendencia hoy día es usar LMS basados en la web, descentralizados y colaborativos (como el famoso Moodle). Aunque, sin lugar a dudas, los LMS no sólo son utilizados en la enseñanza o ámbitos universitarios, sino que se pueden (y de hecho es algo común) utilizar en el mundo empresarial o en el entrenamiento de determinados grupo social [74].

Durante la revisión sistemática de la literatura, se encontró un concepto que merece la pena mencionar. Es lo que el autor llamó Sistemas e-learning Adaptativos (en inglés *Adaptive E-Learning Systems*, o su acrónimo AES). En [106] se argumenta que un LMS tradicional es estático y ofrece a los usuarios los mismos servicios y contenidos, de modo que todos ellos reciben el mismo curso basado en LMS, independientemente de su conocimiento, objetivos e intereses, con acceso al mismo material y mismas herramientas, sin ningún tipo de soporte personalizado. La adaptación y personalización recibe muy poca cobertura en estas plataformas de e-learning. Un curso mediante e-learning no debería ser diseñado sin la unión de las necesidades de estudiantes y profesores y sin adaptaciones durante el progreso y avance del curso. Es por ello que AES se presenta como un tipo de sistemas web adaptativos que intenta ser un acercamiento al e-learning personalizado. La caracterización, la adaptación de contenidos y el aprendizaje colaborativo son algunas de las características que provee AES.

En el artículo [106] puede encontrarse el amplio estudio que compara LMS y AES, usando herramientas *Open Source* para construir sistemas personalizados y adaptativos.

Este enfoque de sistemas adaptativos puede encontrarse en otros autores, como [87], [106] o [20].

5.4.4. PI4: ¿Qué aplicación tiene MDE (o MDA) en LMS?

Tras las preguntas anteriores, el Trabajo de Investigación tenía como objetivo principal la búsqueda de la aplicación de MDE o MDA en LMS.

La libertad que ofrece MDE o MDA para el desarrollo de un LMS queda patente en todos los artículos que se han encontrado, como en [61] o [72]. Existen infinidad de necesidades puesto que no hay dos cursos o sistemas iguales. Es por ello que se requiere diseñar un DSL para cada uno de ellos. Y MDA o MDE es la herramienta que provee los mecanismos para crear los LMS.

Como se pudo leer en el capítulo 2, tanto MDA como MDE ayudan al diseño de sistemas TEL y a la generación de un LMS, potenciando y ayudando a:

Crear lenguajes de modelado específico que aseguren tanto la expresividad contextual como la pedagógica. Estos lenguajes permiten descripciones más precisas con un detalle semántico no alcanzable con modelos genéricos.

Crear herramientas de diseño específico basadas en estándares y creación de generadores como EMF o GMF de Eclipse.

Reutilizar modelos de escenarios de aprendizaje a través de estándares como MOF (*Meta Object Facility*).

Desarrollar potencialmente cualquier LMS que es el mayor beneficio ofrecido por MDE y MDA.

Desplegar automáticamente sistemas TEL en el LMS escogido o desarrollado. Los diseñadores pueden enfocarse en precisamente eso, diseñar, en lugar de lidiar con dificultades técnicas.

Capítulo 6

Conclusiones y trabajos futuros

A través del Trabajo de Investigación presentado y desarrollado, se ha pretendido abarcar principalmente dos objetivos: por un lado, ahondar en los conceptos que, hasta entonces, eran desconocidos para mí, y por otro, realizar una revisión de la literatura, siguiendo un proceso claro y basado en [51] y [11].

El primer objetivo permite apreciar claramente que los DSL son el camino a seguir y por el cual hay que continuar investigando, con la intención de mejorar lo ya existente en los LMS. Más concretamente, MDE tiene todo el potencial necesario para seguir evolucionando las plataformas de e-learning, asunto que ha sido tratado en la sección 4.2.

El segundo objetivo es, sin duda, vital para conocer el “terreno” por el que nos movemos. Se extrae de los artículos y estudios analizados la íntima relación entre MDE, MDA y los LMS. Hay numerosas tecnologías software y metodologías que aparecen continuamente con el ánimo de ir avanzando en las plataformas actuales. Desde los nombres más conocidos (como Moodle o Claroline) hasta invenciones propias y prometedoras que pueden encontrarse en la literatura recogida en este trabajo.

Personalmente, me hubiera gustado profundizar más y detallar en mayor medida cada uno de los elementos tratados. Y de hecho, el trabajo futuro pasa por enfocar las investigaciones hacia MDE y los LMS adaptativos, consiguiendo así reducir el campo de estudio. Si nos basamos en los artículos recogidos y en el análisis realizado, se puede afirmar que MDE es aplicable en el contexto de diseño de cursos para LMSs y que reduce significativamente el tiempo y esfuerzo en la construcción y despliegue de cursos sobre plataformas LMS.

Antes de cerrar este capítulo, hay que añadir que este Trabajo de Investigación ha servido, entre otras muchas cosas, para ampliar conocimientos en la gestión de documentación, gestión de referencias, búsqueda de documentación, artículos o estudios, así como experimentar con todo tipo de librerías digitales y búsquedas con el objetivo de encontrar la información necesaria. No obstante,

se han encontrado algunos pasos que por su idiosincrasia o naturaleza merecen ser tratados aquí, pues la experiencia adquirida para estas fases deberían ayudar al lector o a futuros investigadores.

Se quiere por tanto hablar de los siguientes puntos y contar la experiencia propia para cada uno de ellos.

Motores de búsqueda Antes de iniciar una revisión sistemática de la literatura, es importante tener en cuenta la adecuación de las búsquedas a cada motor, ya que no en todos sitios las cadenas de búsqueda se utilizan por igual. Por ejemplo, en *Science Direct* hay menos campos por los que buscar que en *IEEEExplore*, o dependiendo de si la búsqueda es sencilla o avanzada, en según qué motores se da más importancia a las palabras clave o a la búsqueda en todo el documento.

Resultados relevantes Se necesita tener muy presente la forma de filtrar los resultados relevantes antes de iniciar las búsquedas. Suele ocurrir que, una vez definidas las cadenas, se obtienen más de 100 o 200 resultados. Lógicamente, no siempre se puede estudiar tal cantidad de documentos. Quizá la cadena de búsqueda utilizada no es la correcta, pero si lo fuera, serían necesarias herramientas externas para analizar el fichero objetivo y descartar o no en base a una serie de patrones o reglas.

Procesamiento de la salida Ha sido tedioso organizar toda la información recabada. En el caso de este trabajo, se han usado una serie de hojas de cálculo para almacenar los estudios, darles unos identificadores únicos y poder trabajar de forma cómoda con ellos. Esto ha permitido generar tablas como las presentadas en el documento. Sin embargo, lo ideal hubiera sido disponer de algún tipo de software que, recogiendo el resultado de las búsquedas, guardara esa información en un formato (XML, quizá, o tal vez BibTEX) para poder importar y exportar fácilmente los resultados.

Trabajar con referencias Es imprescindible tener bien organizadas las referencias. Aquí se ha optado por BibTEX y el gestor Mendeley junto a JabRef, este último muy útil para exportar bloques de referencias a formato BibTEX. Pero todo esto no evita que haya que copiar esas referencias a mano a partir de los estudios recogidos. Es por ello que si se tuviera un formato estándar (como se comentaba en el punto anterior), sería mucho más eficiente el pasar las referencias a L^AT_EX 2_ε. De otro modo, el proceso es sin lugar a dudas costoso en tiempo. Una vez tenemos los documentos, hay que incorporarlos al gestor de referencias y completar campos como año, *journal*, el título correctamente, etc., que son cosas que no se rellenan siempre automáticamente tras la carga del correspondiente PDF en el software usado. O incluso que sí se rellenen automáticamente, pero mal, con títulos cortados o números de página que no son los correctos.

Grado de confianza Los resultados que se recogen de los motores de búsqueda no indican cuánto de confiables son. Es decir, no se sabe a priori si eso que

se obtiene “vale” mucho o poco. A la hora de realizar una investigación, sería interesante asignar un grado de confianza al resultado, sobre todo cuando las cadenas contienen un “o” lógico (*OR*). Si el estudio obtenido contiene todas las palabras clave o *keywords* es mejor resultado que si sólo contiene una o dos palabras clave. Esto debería poder servir para descartar resultados.

Calidad de los estudios Como se vió en la sección 5.2.3, se puntuó cada estudio para quedarnos con los que cumplían un cierto nivel. Esto, actualmente, es totalmente manual. Usando las hojas de cálculo, se fue otorgando una puntuación según lo que se iba estudiando de cada artículo. A pesar de que esto es difícil de automatizar, sí que ayudaría tener alguna herramienta con la que poder medir la calidad de cada estudio puntuándolos en base a preguntas definidas por el usuario. Por ejemplo: ¿está el artículo en inglés?, ¿el artículo o estudio tiene un apartado de conclusiones?, ¿es claro y conciso el objetivo que busca el estudio? Pero como se acaba de mencionar, se antoja complicado evitar la intervención del ser humano.

Extracción de datos En la sección 5.3 se extrajo información de cada estudio. Al igual que con el aseguramiento de la calidad, esto tiene que hacerlo inevitablemente el humano, pero sabiendo que existen una serie de palabras clave o *keywords*, disponer de herramientas automáticas que extraigan para cada artículo esas palabras clave sería una primera clasificación muy útil en esta fase de análisis y extracción de datos.

Capítulo 7

Conceptos

7.1. Glosario genérico

Cuando se trata de ahondar en la ingeniería dirigida por modelos y unir esto al aprendizaje basado en la tecnología (TEL), hay numerosos conceptos que conviene conocer. A continuación se listan los más importantes:

ADDS Approach to Document-Based Development of Software.

CMS Content Management Systems.

DSL Domain Specific Language. Lenguaje de programación orientado a un problema específico.

DSML Domain Specific Modeling Languages.

ECM Enterprise Content Management.

EML Educational Modeling Language. El modelado de procesos educativos es un aspecto clave en la construcción de aplicaciones de aprendizaje más efectivas [44]. Los EML utilizados pueden ser uno de los estándares ya existentes (por ejemplo, IMS Learning Design o UML), la personalización de un estándar para llegar a un perfil de aplicación determinado, o incluso un EML de dominio específico especialmente diseñado para encajar mejor en un escenario de aprendizaje con necesidades muy particulares.

GUI Graphical User Interface.

IMS IMS-Learning Design (LD). Básicamente, Learning Design (IMS-LD) es una herramienta de modelado que utiliza la metáfora del juego teatral para describir un proceso de enseñanza y aprendizaje [57]. Sus principales componentes son: metadatos, roles, actividades, entorno, qué papel toma el rol (por ejemplo, actividades del actor, quién hace qué, cuándo y cómo), secuencia de actividades dentro del papel que toma el rol, condiciones y

notificaciones (interactividad y control sobre un diseño de aprendizaje vivo como forma de un sistema de mensajes dirigidos por eventos). A través de la herramienta IMS-LD uno puede formalmente expresar una unidad de aprendizaje (*UoL*, en inglés).

IMS LTI Learning Tools Interoperability.

LMS Learning Management System. Estos sistemas pueden dividirse en dos categorías principales: iniciativas de código abierto como Moodle, Sakai, ATutor, y Whiteboard; y soluciones propietarias que incluyen WebCT/Blackboard, Gradepoint, Desire2Learn, y Learn.com.

LTSA Learning Technology Systems Architecture. En términos de estándar para tecnologías de aprendizaje, LTSA especifica una arquitectura de alto nivel para los sistemas de aprendizaje, educación y entrenamiento basados en la tecnología, describiendo el diseño de alto nivel y los componentes de estos sistemas [110].

M2M Model 2 Model. Herramientas utilizadas: SmartQVT, Operational QVT, Declarative QVT o ATL.

M2T Model 2 Text. Herramientas utilizadas: MOFScript, JET, Acceleo, Xpand o Xtend.

MDA Model-Driven Architecture. Podríamos decir que MDA equivale a MDE. Lanzado por el *Object Management Group* (OMG) en 2001 como *trade-mark*. Por ello, MDA es la “visión particular” de MDE por parte del OMG.

MDD Model-Driven Development.

MDE Model-Driven Engineering.

MDT Model-Driven Testing. Orientado a la construcción de casos de prueba.

MDWE Model-Driven Web Engineering. Generación automática de modelos de ingeniería (requisitos, análisis, etc.)

MIC Model-Integrated Computing. MIC es una acercamiento al desarrollo software de dominio específico y dirigido por modelos. Utiliza modelos y transformaciones sobre modelos donde los modelos son sentencias de lenguajes de modelado de dominio específico (en inglés, *domain-specific modeling languages*, o su acrónimo DSML). Puede encontrarse mucha más información en [47].

MOF Meta-Object Facility. Muy resumidamente, MOF es un estándar de OMG para la ingeniería dirigida por modelos; un *framework* para la gestión de metadatos. Se utiliza, entre otras cosas, para construir meta-modelos, meta-modelos y modelos. Un número de tecnologías estandarizadas por OMG, incluyendo UML, MOF, CWM, SPEM, XMI y varios perfiles UML, utilizan MOF y tecnologías derivadas de MOF para manipulación de metadatos e intercambio de metadatos-dirigidos [77].

MSF Microsoft Solutions Framework.

RCP Rich Client Platform. Básicamente, es una herramienta que facilita la integración componentes software independientes, donde la mayoría del procesamiento de los datos se produce en el lado del cliente.

RUP Rational Unified Process.

SCORM Sharable Content Object Reference Model.

Muy resumidamente, SCORM es un conjunto de estándares técnicos para productos software relacionados con e-learning. SCORM le “dice” al programador cómo escribir su código de modo que pueda entenderse y manejarse bien con otros software de e-learning. Se puede decir que es el estándar en la industria para la interoperabilidad del software e-learning. Más específicamente, SCORM gobierna cómo se relaciona y “comunica” el contenido online relacionado con el aprendizaje y los sistemas de gestión del aprendizaje (LMS). SCORM no trata sobre el diseño instructivo o cualquier otro concerniente al ámbito pedagógico. Es puramente un estándar técnico [95].

SOA Service Oriented Architecture.

SPE Software Process Engineering. Soporte a la gestión de procesos (metodologías) organizacionales.

TEL Technology-Enhanced Learning. Podemos definir TEL, o mejor dicho, un sistema de este tipo, como un conjunto de componentes heterogéneos e interdependientes (p.e. actores, herramientas y elementos de aprendizaje) organizado en el espacio y el tiempo para satisfacer un objetivo que esté orientado al aprendizaje [27].

UoL Unit of Learning. Una unidad de educación o unidad de entrenamiento autocontenida [57], como pudiera ser un curso, un módulo, una lección, etc.

XMI XML Metadata Interchange. Estándar de OMG para intercambio de información mediante XML.

XML Extensible Markup Language.

7.2. Glosario específico de Eclipse

Eclipse es un entorno de desarrollo software multi-lenguaje [111], con un sistema de *plug-ins* extensible.

Es Eclipse el entorno que se ha utilizado a modo de “laboratorio” para experimentar con las tecnologías que rodean MDE. De hecho, y no en vano, el ecosistema de Eclipse [113], junto con el OMG, son dos de las iniciativas más conocidas de MDE, y todo gracias a su extenso abanico de herramientas de modelado y programación.

CDO Connected Data Objects.

EMF Eclipse Modeling Framework.

GMP Graphical Modeling Project.

TMF Textual Modeling Framework.

XSD XML Schema Definition.

Bibliografía

- [1] Pekka Aho, Janne Merilinna, and Eila Ovaska. Model-Driven Open Source Software Development - The Open Models Approach. *2009 Fourth International Conference on Software Engineering Advances*, pages 185–190, September 2009.
- [2] Marc El Alami, N Casel, and D Zampunieris. An Architecture for e-Learning System with Computational Intelligence. *Knowledge-Based Intelligent Information and Engineering Systems*, 4693:58–65, 2007.
- [3] Colin Allison, Stuart Purdie, and Rosa Michaelson. A Grid Based Open Framework for Technology Enhanced Learning. *2008 Eighth IEEE International Conference on Advanced Learning Technologies*, pages 56–60, 2008.
- [4] L. Anido, J. Rodríguez, M. Caeiro, and J.M. Santos. High-level brokerage services for the e-learning domain. *Computer Standards & Interfaces*, 25(4):303–327, August 2003.
- [5] M Asadi, N Esfahani, and R Ramsin. Process patterns for MDA-based software development. *2010 Eighth ACIS International Conference on Software Engineering Research, Management and Applications*, pages 121–128, 2010.
- [6] Mohsen Asadi, Mahdy Ravakhah, and Raman Ramsin. An MDA-Based System Development Lifecycle. *2008 Second Asia International Conference on Modelling & Simulation (AMS)*, pages 836–842, May 2008.
- [7] Paris Avgeriou, S Retalis, and M Skordalakis. Building Quality into Learning Management Systems - An Architecture - Centric Approach. *Advanced Conceptual Modeling Techniques*, 2784:312–324, 2003.
- [8] CC Aydin and G Tirkes. Open source learning management systems in e-learning and Moodle. *Education Engineering (EDUCON), 2010 IEEE*, pages 593 – 600, 2010.
- [9] K. Balasubramanian and A. Gokhale. Developing applications using model-driven design environments. *Computer*, 39(2):33–40, February 2006.

- [10] J Bézivin. In search of a basic principle for model driven engineering. *Novatica Journal, Special Issue*, V(2):1–5, 2004.
- [11] Jorge Biolchini, Paula Gomes Mian, Ana Candida, and Cruz Natali. Systematic Review in Software Engineering. *PESC*, May 2005.
- [12] Tom Boyle and Andrew Ravenscroft. Context and deep learning design. *Computers & Education*, 59(4):1224–1233, December 2012.
- [13] E Breton and J Bézivin. Model driven process engineering. *Computer Software and Applications Conference, 2001. COMPSAC 2001. 25th Annual International*, pages 225 –230, 2001.
- [14] Arnaud Brossard, Mourad Abed, and Christophe Kolski. Taking context into account in conceptual models using a Model Driven Engineering approach. *Information and Software Technology*, 53(12):1349–1369, December 2011.
- [15] Santi Caballé and F Xhafa. Computational Intelligence Infrastructure in Support for Complex e-Learning Systems. *Computational Intelligence for Technology Enhanced Learning*, 273:143–168, 2010.
- [16] Tihomir Calic, Sergiu Dascalu, and Dwight Egbert. Tools for MDA Software Development: Evaluation Criteria and Set of Desirable Features. *Fifth International Conference on Information Technology: New Generations (itng 2008)*, pages 44–50, April 2008.
- [17] Vasian Cepa and Mira Mezini. Language support for model-driven software development. *Science of Computer Programming*, 73(1):13–25, September 2008.
- [18] Xiao Cong, H Zhang, D Zhou, P Lu, and L Qin. A model-driven architecture approach for developing e-learning platform. *Entertainment for Education. Digital Techniques and Systems*, 6249:111–122, 2010.
- [19] RJE Dagdag. Teacher tool for visualization and management of a technology-enhanced learning environment. *TENCON 2011 - 2011 IEEE Region 10 Conference*, pages 1426–1430, 2011.
- [20] Declan Dagger, Alexander O Connor, Séamus Lawless, Eddie Walsh, and Vincent P Wade. Service-Oriented E-Learning Platforms. From Monolithic Systems to Flexible Services. *IEEE Computer Society*, pages 28–35, 2007.
- [21] Marijana Despotović-Zrakić, Dušan Barać, Zorica Bogdanović, Branislav Jovanić, and Božidar Radenković. Integration of web based environment for learning discrete simulation in e-learning system. *Simulation Modelling Practice and Theory*, 27:17–30, September 2012.

- [22] Arie Van Deursen, Paul Klint, and Joost Visser. Domain-specific languages: an annotated bibliography. *ACM Sigplan Notices*, 35(6):26–36, 2000.
- [23] Vladan Devedzic, Jelena Jovanovic, and Dragan Gasevic. The Pragmatics of Current E-Learning Standards. *IEEE Internet Computing*, may-june:19–27, 2007.
- [24] Roberto Di Cosmo, Davide Di Ruscio, Patrizio Pelliccione, Alfonso Pierantonio, and Stefano Zacchiroli. Supporting software evolution in component-based FOSS systems. *Science of Computer Programming*, 76(12):1144–1160, December 2011.
- [25] Juan Manuel Dodero, Álvaro Martínez del Val, and Jorge Torres. An extensible approach to visually editing adaptive learning activities and designs based on services. *Journal of Visual Languages & Computing*, 21(6):332–346, December 2010.
- [26] Juan Manuel Dodero, Álvaro Martínez Del Val, and Iván Ruiz-Rube. FlexoLD: Un Lenguaje Específico de Dominio para Diseños de Aprendizaje. *Actas de los Talleres JISBD*, 4(3):88–97, 2010.
- [27] Rim Drira, Mona Laroussi, Xavier La Pallec, and Bruno Warin. Contextualizing learning scenarios according to different Learning Management Systems. *Learning Technologies, IEEE Transactions on*, 5(3):213–225, 2011.
- [28] Eclipse. Eclipse documentation. <http://www.eclipse.org/documentation>, 2012. [Online; accessed 24-September-2012].
- [29] Jean-Marie Favre and Tam NGuyen. Towards a Megamodel to Model Software Evolution Through Transformations. *Electronic Notes in Theoretical Computer Science*, 127(3):59–74, April 2005.
- [30] Liliana Favre and Liliana Martinez. Formalizing MDA components. *Reuse of Off-the-Shelf Components*, 4039(Cic):326–339, 2006.
- [31] MA Forment, X Pedro, and MJ Casañ. Learning Assessment Using Wikis: Integrated or LMS Independent? *Knowledge Management, Information Systems, E-Learning, and Sustainability Research*, pages 150–158, 2010.
- [32] Robert France and Bernhard Rumpe. Model-driven Development of Complex Software : A Research Roadmap Model-driven Development of Complex Software : A Research Roadmap. *Future of Software Engineering, 2007. FOSE '07*, pages 37 – 54, 2007.
- [33] José Luis Garrido, Manuel Noguera, Miguel González, María V. Hurtado, and María L. Rodríguez. Definition and use of Computation Independent Models in an MDA-based groupware development process. *Science of Computer Programming*, 66(1):25–43, April 2007.

- [34] José M. Gascueña, Elena Navarro, and Antonio Fernández-Caballero. Model-driven engineering techniques for the development of multi-agent systems. *Engineering Applications of Artificial Intelligence*, 25(1):159–173, February 2012.
- [35] Tahar Gherbi, D Meslati, and I Borne. MDE between Promises and Challenges. *Computer Modelling and Simulation, 2009. UKSIM '09*, pages 152–155, 2009.
- [36] Mehdi Fahmideh Gholami and Raman Ramsin. Strategies for Improving MDA-Based Development Processes. *2010 International Conference on Intelligent Systems, Modelling and Simulation*, pages 152–157, January 2010.
- [37] Miguel Ángel Conde González, Francisco José García Peñalvo, María José Casany Guerrero, and Marc Alíer Forment. Adapting LMS Architecture to the SOA: An Architectural Approach. *2009 Fourth International Conference on Internet and Web Applications and Services*, pages 322–327, 2009.
- [38] Cesar Gonzalez-Perez and Brian Henderson-Sellers. Modelling software development methodologies: A conceptual foundation. *Journal of Systems and Software*, 80(11):1778–1796, November 2007.
- [39] Bas Graaf. Model-Driven Evolution of Software Architectures. *11th European Conference on Software Maintenance and Reengineering (CSMR'07)*, pages 357–360, 2007.
- [40] Reyes Grangel, R Chalmeta, and C Campos. A Proposal for Goal Modelling Using a UML Profile. *Enterprise Interoperability III*, pages 679–690, 2008.
- [41] Abdelwahab Hamou-Lhadj, Abdelouahed Gherbi, and Jagadeesh Nandigam. The Impact of the Model-Driven Approach to Software Engineering on Software Engineering Education. *2009 Sixth International Conference on Information Technology: New Generations*, pages 719–724, 2009.
- [42] Davinia Hernández-leo, Andreas Harrer, Juan Manuel Dodero, Juan I Asensio-pérez, and Daniel Burgos. A Framework for the Conceptualization of Approaches to Create-by-Reuse of Learning Design Solutions. *Journal of Universal Computer Science*, 13(7):750–760, 2007.
- [43] John Hutchinson, Jon Whittle, Mark Rouncefield, and Steinar Kristoffersen. Empirical assessment of MDE in industry. *Proceeding of the 33rd international conference on Software engineering - ICSE '11*, page 471, 2011.
- [44] Zhi Jiang. The educational modeling languages in Instructional Design: Towards a UML applications. *2011 6th International Conference on Computer Science & Education (ICCSE)*, pages 162–165, August 2011.

- [45] Lynne Jump. Why university lecturers enhance their teaching through the use of technology: a systematic review. *Learning, Media and Technology*, 36(1):55–68, March 2011.
- [46] JE Kareinen and J Pötry. Enterprise Portal Technology in Computer-Supported Collaborative Learning. *The Impact of Virtual, Remote, and Real Logistics Labs*, 282:54–61, 2012.
- [47] Gabor Karsai and Aditya Agrawal. Graph transformations in OMG’s model-driven architecture. *Applications of Graph Transformations with Industrial Relevance*, pages 243–259, 2004.
- [48] Gabor Karsai, Sandeep Neema, and David Sharp. Model-driven architecture for embedded software: A synopsis and an example. *Science of Computer Programming*, 73(1):26–38, September 2008.
- [49] Jared Keengwe and D Georgina. The digital course training workshop for online learning and teaching. *Education and Information Technologies*, 17(4):365–379, 2012.
- [50] Stuart Kent. Model Driven Engineering, 2002.
- [51] B. Kitchenham. Guidelines for performing Systematic Literature Reviews in Software Engineering. *EDSE Technical Report*, 2007.
- [52] V Kulkarni and S Reddy. A model-driven architectural framework for integration-capable enterprise application product lines. *Model Driven Architecture-Foundations and Applications*, 4066:1–12, 2006.
- [53] Pierre Laforcade. A Domain-Specific Modeling approach for supporting the specification of Visual Instructional Design Languages and the building of dedicated editors. *Journal of Visual Languages & Computing*, 21(6):347–358, December 2010.
- [54] D.H. Leo. IMS learning design support for the formalization of collaborative learning patterns. *Advanced Learning Technologies, 2004. Proceedings. IEEE .*, pages 350 – 354, 2004.
- [55] Grzegorz Loniewski. An architecture-oriented model-driven requirements engineering approach. *Model-Driven Requirements Engineering Workshop (MoDRE), 2011*, pages 31–38, 2011.
- [56] Francisco J. Lucas, Fernando Molina, and Ambrosio Toval. A systematic review of UML model consistency management. *Information and Software Technology*, 51(12):1631–1645, December 2009.
- [57] I. Magnisalis and S. Demetriadis. Extending IMS-LD Capabilities: A Review, a Proposed Framework and Implementation Cases. In Thanasis Daradoumis, Stavros N. Demetriadis, and Fatos Xhafa, editors, *Intelligent*

Adaptation and Personalization Techniques in Computer-Supported Collaborative Learning, volume 408 of *Studies in Computational Intelligence*, pages 85–108. Springer Berlin / Heidelberg, 2012.

- [58] I Magnisalis and S Demetriadis. Extending IMS-LD Capabilities: A Review, a Proposed Framework and Implementation Cases. *Intelligent Adaptation and Personalization Techniques in Computer-Supported Collaborative Learning*, 408:85–108, 2012.
- [59] I. Magnisalis, S. Demetriadis, and A. Karakostas. Adaptive and Intelligent Systems for Collaborative Learning Support: A Review of the Field. *IEEE Transactions on Learning Technologies*, 4(1):5–20, January 2011.
- [60] O.S. Marín, C.E.M. and García, P.A.G. and Lovelle, J.M.C. and Martínez. Application of model-driven engineering (MDA) for the construction of a tool for domain-specific modeling (DSM) and the creation of modules in learning management systems (LMS) platform independent. *DYNA*, 78(169):43–52, 2011.
- [61] I Martinez-Ortiz. Authoring and reengineering of IMS Learning Design units of learning. *Learning Technologies, IEEE Transactions on*, 2(3):189–202, 2009.
- [62] Iván Martínez-Ortiz, José Luis Sierra, and Baltasar Fernández-Manjón. Enhancing IMS LD Units of Learning Comprehension. *2009 Fourth International Conference on Internet and Web Applications and Services*, pages 561–566, 2009.
- [63] Iván Martínez-Ortiz, José-Luis Sierra, Baltasar Fernández-Manjón, and Alfredo Fernández-Valmayor. Language engineering techniques for the development of e-learning applications. *Journal of Network and Computer Applications*, 32(5):1092–1105, September 2009.
- [64] Nicholas Mavengere and M Ruohonen. Using Open Source Software for Improving Dialog in Computer Science Education - Case Mozambique University. *Information Technology and Managing Quality Education*, 348:52–61, 2011.
- [65] E Michael Maximilien, Hernan Wilkinson, Nirmit Desai, and Stefan Tai. A Domain-Specific Language for Web APIs and Services Mashups. *ServiceOriented Computing ICSOC 2007*, 4749:13–26, 2007.
- [66] Jose-Norberto Mazón and Juan Trujillo. An MDA approach for the development of data warehouses. *Decision Support Systems*, 45(1):41–58, 2008.
- [67] Marjan Mernik, Jan Heering, and Anthony M. Sloane. When and how to develop domain-specific languages. *ACM Computing Surveys*, 37(4):316–344, December 2005.

- [68] M Minovic, M Milovanovic, and D Starcevic. Using Learning Objects in Games. *Knowledge Management, Information Systems, E-Learning, and Sustainability Research*, 111:297–305, 2010.
- [69] M. Minovic, M. Milovanovic, and D. Starcevic. Learning object repurposing for various multimedia platforms. *Multimedia Tools and Applications*, pages 1–20, 2011.
- [70] Ana I. Molina, William J. Giraldo, Jesús Gallardo, Miguel a. Redondo, Manuel Ortega, and Guillermo García. CIAT-GUI: A MDE-compliant environment for developing Graphical User Interfaces of information systems. *Advances in Engineering Software*, 52:10–29, October 2012.
- [71] Carlo Montangero and Laura Semini. Barbed Model-Driven Software Development: A Case Study. *Electronic Notes in Theoretical Computer Science*, 207:171–186, April 2008.
- [72] CE Montenegro-Marín. Domain specific language for the generation of learning management systems modules. *Journal of Web Engineering*, 11(1):23–50, 2012.
- [73] Yishay Mor and Niall Winters. Design approaches in technology-enhanced learning. *Interactive Learning Environments*, 2007.
- [74] Jad Najjar and T Klobucar. Towards outcome based learning: An engineering education case. *Global Engineering Education Conference (EDU-CON)*, 2011 IEEE, pages 1039–1048, 2011.
- [75] Vladimirs Nikulsins and Oksana Nikiforova. Adapting Software Development Process towards the Model Driven Architecture. *2008 The Third International Conference on Software Engineering Advances*, pages 394–399, October 2008.
- [76] OMG. MDA Guide Version 1.0.1. <http://www.omg.org/cgi-bin/doc?omg/03-06-01>, June 2003. [Online; accessed 17-September-2012].
- [77] OMG. Meta Object Facility (MOF) Core Specification. <http://www.omg.org/spec/MOF/2.0/>, January 2006. [Online; accessed 17-September-2012].
- [78] Claus Pahl. Semantic model-driven architecting of service-based software systems. *Information and Software Technology*, 49(8):838–850, August 2007.
- [79] Daniel Perovich, Maria Cecilia Bastarrica, and Cristian Rojas. Model-Driven approach to Software Architecture design. *2009 ICSE Workshop on Sharing and Reusing Architectural Knowledge*, pages 1–8, May 2009.
- [80] Francisco J. Pino, Félix García, and Mario Piattini. Software process improvement in small and medium software enterprises: a systematic review. *Software Quality Journal*, 16(2):237–261, Nov 2007.

- [81] Nicolas Prat, Jacky Akoka, and Isabelle Comyn-Wattiau. An MDA approach to knowledge engineering. *Expert Systems with Applications*, 39(12):10420–10437, September 2012.
- [82] L Prieto and J Asensio-Pérez. GLUE!-PS: A multi-language architecture and data model to deploy TEL designs to multiple learning environments. *Towards Ubiquitous Learning*, 6964:285–298, 2011.
- [83] Mirja Pulkkinen, D Kozlov, and J Pawlowski. Software and Standards in an Emerging Domain. *Software Business*, 51:217–222, 2010.
- [84] Maja Pusnik, B Sumak, and M Hericko. Investigation of Virtual Learning Environment in the Context of Web 2.0. *Mobile, Hybrid, and On-Line Learning, 2010. ELML '10*, pages 1–6, 2010.
- [85] Outi Räihä. A survey on search-based software design. *Computer Science Review*, 4(4):203–249, November 2010.
- [86] Adrian Rawlings and Peter Van Rosmalen. Survey of Educational Modelling Languages (EMLs). *CEN/ISSS WS/LT Learning Technologies Workshop*, 2002.
- [87] N Ruiz Reyes. Comparing open-source e-learning platforms from adaptivity point of view. *EAEEIE Annual Conference, 2009*, pages 1–6, 2009.
- [88] D. Sampson and D. Fytros. Competence models in technology-enhanced competence-based learning. *Handbook on Information Technologies for Education and Training*, pages 155–177, 2008.
- [89] Demetrios Sampson, K Gotze, and Panayiotis Zervas. Delivering IMS Learning Design Activities via Mobile Devices. *Advanced Learning Technologies, 2007. ICAIT 2007.*, pages 367–368, July 2007.
- [90] Demetrios G. Sampson, Panagiotis Zervas, and Sofoklis Sotiriou. Open Access to Science Education Resources and Learning Designs in Europe. *2012 IEEE Fourth International Conference on Technology for Education*, pages 200–203, July 2012.
- [91] Iván Santiago, Álvaro Jiménez, Juan Manuel Vara, Valeria De Castro, Verónica a. Bollati, and Esperanza Marcos. Model-Driven Engineering as a new landscape for traceability management: A systematic literature review. *Information and Software Technology*, 54(12):1340–1356, December 2012.
- [92] João De Sousa Saraiva and Alberto Rodrigues Da Silva. CMS-Based Web-Application Development Using Model-Driven Languages. *2009 Fourth International Conference on Software Engineering Advances*, pages 21–26, September 2009.

- [93] Eileen Scanlon. Technology Enhanced Informal Science Learning and Engagements: A Typology of Activity. *2012 IEEE 12th International Conference on Advanced Learning Technologies*, pages 536–540, July 2012.
- [94] D.C. Schmidt. Model-Driven Engineering. *IEEE Computer*, 39 (2):25–31, 2006.
- [95] Scorm. SCORM Explained. <http://www.scorm.com/scorm-explained/>, 2012. [Online; accessed 17-November-2012].
- [96] Gayane Sedrakyan and Monique Snoeck. Technology-Enhanced Support for Learning Conceptual Modeling. *Enterprise, Business-Process and Information Systems Modeling*, 113:435–449, 2012.
- [97] B. Selic. The pragmatics of model-driven development. *IEEE Software*, 20(5):19–25, September 2003.
- [98] Sam Sengupta, Geethapriya Thamilarasu, Akshay Dalvi, and Shalini Vellanki. ACM-learning architecture: A normative proposal for technology enhanced education for developing countries. *2012 IEEE International Conference on Technology Enhanced Education (ICTEE)*, pages 1–5, January 2012.
- [99] J Sierra and A Fernández-Valmayor. Building applications with domain-specific markup languages: a systematic approach to the development of XML-based software. *Web Engineering*, pages 230–240, 2003.
- [100] M.A.G. Silva. Model-driven development of learning objects. *Frontiers in Education Conference (FIE)*, 2011, pages F4E–1 – F4E–6, 2011.
- [101] Yashwant Singh and Manu Sood. Models and Transformations in MDA. *2009 First International Conference on Computational Intelligence, Communication Systems and Networks*, pages 253–258, July 2009.
- [102] Pierre Tchounikine, AI Mørch, and LJ Bannon. A Computer Science Perspective on Technology-Enhanced Learning Research. *Technology-Enhanced Learning*, 2009.
- [103] Laurent Thiry and Bernard Thirion. Functional metamodels for systems and software. *Journal of Systems and Software*, 82(7):1125–1136, July 2009.
- [104] Victoria Torres, P Giner, and V Pelechano. Developing BP-driven web applications through the use of MDE techniques. *Software and Systems Modeling*, 11(4):609–631, 2010.
- [105] Javier Troya, Antonio Vallecillo, Francisco Durán, and Steffen Zschaler. Model-driven performance analysis of rule-based domain specific visual models. *Information and Software Technology*, 55(1):88–110, January 2013.

- [106] Dimitrios Tsolis, Sofia Stamou, Paraskevi Christia, Soultana Kampana, Triseugeni Rapakoulia, Maria Skouta, and Athanasios Tsakalidis. An adaptive and personalized open source e-learning platform. *Procedia - Social and Behavioral Sciences*, 9:38–43, January 2010.
- [107] Juan Manuel Vara and Esperanza Marcos. A framework for model-driven development of information systems: Technical decisions and lessons learned. *Journal of Systems and Software*, 85(10):2368–2384, October 2012.
- [108] Dessislava Vassileva, Boyan Bontchev, Boryana Chavkova, and Vladimir Mitev. Software Construction of an Authoring Tool for Adaptive E-learning Platforms. *2009 Fourth Balkan Conference in Informatics*, pages 187–192, 2009.
- [109] Lars Vogel. Eclipse Modeling Framework (EMF) – Tutorial. <http://www.vogella.com/articles/EclipseEMF/article.html>, 2012. [Online; accessed 24-September-2012].
- [110] Hong Wang, D Zhang, and J Zhou. MDA-based development of e-learning system. *Computer Software and Applications Conference*, 2003.
- [111] Wikipedia. Eclipse (software). [http://en.wikipedia.org/wiki/Eclipse_\(software\)](http://en.wikipedia.org/wiki/Eclipse_(software)), 2012. [Online; accessed 6-September-2012].
- [112] Wikipedia. Learning management system. http://en.wikipedia.org/wiki/Learning_management_system, 2012. [Online; accessed 15-December-2012].
- [113] Wikipedia. Model-driven engineering. http://en.wikipedia.org/wiki/Model-driven_engineering, 2012. [Online; accessed 6-September-2012].
- [114] Wikipedia. Technology-Enhanced Learning. http://en.wikipedia.org/wiki/Technology-Enhanced_Learning, 2012. [Online; accessed 3-October-2012].
- [115] L. Zhu. Model-driven architecture. *Essential software architecture*, pages 197–215, 2006.

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

`<http://fsf.org/>`

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein.

The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such

following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “**Entitled XYZ**” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document’s license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover

Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.